



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

ULB

## Logics with Invariantly Used Relations

Eickmeyer, Kord

(2020)

DOI (TUprints): <https://doi.org/10.25534/tuprints-00013503>

Lizenz:



CC-BY 4.0 International - Creative Commons, Attribution

Publikationstyp: Habilitation

Fachbereich: 04 Department of Mathematics

Quelle des Originals: <https://tuprints.ulb.tu-darmstadt.de/13503>

---



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Mathematik

# Logics with Invariantly Used Relations

---

vom Fachbereich Mathematik der Technischen Universität Darmstadt  
genehmigte Habilitationsschrift  
von **Dr. Kord Eickmeyer** aus Lage

---

Datum des Habilitationsvortrags: 13. Dezember 2019

- 1. Gutachter: Prof. Dr. Martin Otto
- 2. Gutachter: Prof. Dr. Marc Pfetsch
- 3. Gutachter: Prof. Dr. Rodney Downey
- 4. Gutachter: Prof. Dr. Mikołaj Bojańczyk

# **Logics with Invariantly Used Relations**

Habilitationsschrift

Dr. Kord Eickmeyer

# Contents

<b>I</b>	<b>Background</b>	<b>11</b>
<b>1</b>	<b>Preliminaries and Notation</b>	<b>13</b>
1.1	Logic Preliminaries . . . . .	14
<b>2</b>	<b>Logics with Invariantly Used Relations</b>	<b>25</b>
2.1	Invariantly Used Relations . . . . .	25
2.2	Undecidability of the Syntax . . . . .	30
<b>3</b>	<b>Structural Properties of Graphs</b>	<b>35</b>
3.1	Trees and Tree Decompositions . . . . .	37
3.2	Treedepth . . . . .	40
3.3	Graphs of Bounded Genus . . . . .	44
3.4	Minors and Topological Subgraphs . . . . .	46
<b>II</b>	<b>Expressivity and Succinctness</b>	<b>51</b>
<b>4</b>	<b>Variants of First-Order Logic</b>	<b>53</b>
4.1	Separation Results . . . . .	54
4.2	Trees . . . . .	55
4.3	Locality . . . . .	59
4.4	Depth-Bounded Structures . . . . .	62
<b>5</b>	<b>Order-Invariant Monadic Second-Order Logic</b>	<b>75</b>
5.1	Depth-Bounded Structures . . . . .	75
5.2	Decomposable Structures . . . . .	80
<b>III</b>	<b>Algorithmic Meta-Theorems</b>	<b>83</b>
<b>6</b>	<b>The Complexity of Model Checking</b>	<b>85</b>

<b>7</b>	<b>Successor-Invariant First-Order Logic</b>	<b>91</b>
7.1	Interpreting a Successor Relation . . . . .	92
7.2	$k$ -Walks in Graphs with Excluded Minors . . . . .	96
7.3	$k$ -Walks in Graphs with Excluded Topological Subgraphs . .	101
7.4	The Model Checking Algorithm . . . . .	107
<b>8</b>	<b>Dense Graphs</b>	<b>109</b>
	<b>Bibliography</b>	<b>112</b>
	<b>Index</b>	<b>122</b>

# Introduction

This thesis deals with various aspects of the finite model theory of logics with invariantly used relations. To construct such a logic we start with an arbitrary logic  $L$ , such as first-order or monadic second-order logic. Sentences in such a logic are statements about structures, i.e. sets with named constants, relations, and functions defined on them. We now enrich the logic  $L$  by giving it the ability to speak about additional relations such as a linear order which is not actually defined on the structure in question, provided that its truth value be independent of which particular linear order we choose.

This may seem like a somewhat contrived set-up, but it pops up naturally in several contexts in finite model theory. One reason for this is that a linear order can be used as a device for symmetry breaking. By Immerman and Vardi's well known result [Imm82, Var82, Liv82], fixed-point logics such as least fixed-point logic (**LFP**) capture polynomial time on ordered structures, while on sets they gain absolutely no expressive power over first-order logic. This is because the computational models used to define polynomial time inherently work on strings, and representing a mathematical structure as a string brings with it some (representation-dependent) linear order on the structure.

Many algorithms tacitly rely on this linear order for symmetry breaking: Gaussian elimination for systems of linear equations has to repeatedly choose variables for pivoting, algorithms for topologically sorting the vertices in a directed acyclic graph need to somehow decide upon the order of source nodes etc. That **LFP** reduces to first-order logic on sets without further structure reflects the fact that being able to iterate can be worthless if there is no order in which to iterate.

One approach to tackling this problem is by taking away this power of symmetry breaking from polynomial time, resulting in formalisms such as choiceless polynomial time [BGS99, DRR08], or by carefully adding ex-

pressive power without enabling symmetry breaking, leading to logics such as fixed-point logics with rank operators [DGHL09, GP15] as candidates for logics with a decidable syntax capturing polynomial time on all finite structures.

A different approach is to explicitly add the power to break symmetries to a logic capturing polynomial time on ordered structures, by letting sentences in this logic speak about a linear order. But how should such a linear order be chosen? On structures with non-trivial automorphisms there is no canonical order (and even if there was one, it is not at all clear that it should be polynomial-time computable). Thus it is natural to restrict attention to sentences whose truth value does not depend on the specific linear order, but only on the fact that it is *some* linear order. These are exactly the kinds of sentences we are studying in this thesis.

The problem with these logics is that they are defined by referring to a semantic condition, namely that of being invariant under the particular choice of linear order. This property is undecidable, and it remains undecidable even when restricted to classes of structures such as strings or star forests, with coloured sets being a notable exception (cf. Section 2.2). However, this does not rule out the possibility of there being another logic with the same expressive power and whose syntax is decidable. Thus in Part II we compare the expressive power of logics with invariantly used relations to that of plain logics (which have a decidable syntax) and obtain collapse results on various classes of structures.

Another area where invariantly used relations present themselves as a natural concept is model checking: Given a finite structure  $A$  and a sentence  $\varphi$ , decide whether  $A \models \varphi$ . This is the decision variant of model checking, other variants allow formulae with free variables instead of just sentences and ask for some variable binding which satisfies the formula (search) or the number of variable bindings that satisfy the formula (counting).

This problem is of practical importance for several reasons: First of all, it is a reasonable abstraction of decision/search/counting problems in database theory, since relational databases can be modelled as relational structures, and the relational core of the database language SQL (Structured Query Language) corresponds to first-order logic. Given that a database is somehow represented in computer memory, it is natural to take the universe of the relational structure modelling the database to be an initial segment  $\{0, \dots, n\}$  of the natural numbers.<sup>1</sup> Two questions immediately arise:

---

<sup>1</sup>Incidentally, SQL evolved out of earlier database query languages such as ISAM or

1. Does the expressive power of our query language increase if we allow access to the index set (e.g. by its linear order, or by saying that the index of some element is the sum of the indices of certain other elements), while requiring the query to be invariant under re-indexings?
2. Does the computational cost of evaluating database queries increase if one allows invariant access to the index set?

Roughly speaking, Part II deals with the first of these questions, while Part III addresses the second one.

Apart from modelling database queries, model checking can be used as a natural generalisation of a wide range of algorithmic problems. For example, a graph contains a clique of size  $k$  if, and only if, it satisfies the sentence

$$\exists x_1 \dots \exists x_k \left( \bigwedge_{i < j} Ex_i x_j \right),$$

it contains a dominating set of size  $k$  if, and only if, it satisfies the sentence

$$\exists x_1 \dots \exists x_k \forall y \left( \bigvee_i (x_i \doteq y \vee Ex_i y) \right),$$

and it is  $k$ -colourable if, and only if, it satisfies the MSO-sentence

$$\exists X_1 \dots \exists X_k \left( \left( \forall x \bigvee_i X_i x \right) \wedge \left( \bigwedge_i \forall x \forall y (X_i x \wedge X_i y \rightarrow \neg Exy) \right) \right).$$

Thus, efficient model checking algorithms also yield efficient algorithms for these (and many other) problems. Note that model checking can not be efficient if running time is measured as a function of both the formula  $\varphi$  and the structure  $A$ , since model checking even for first-order logic is PSPACE-complete already when  $A$  is a two-element set without further structure. The best one can hope for, then, is efficient *parameterised* algorithms, in the sense of *fixed-parameter tractable (fpt)* algorithms, i.e. algorithms with a running time of

$$f(\varphi) \cdot |A|^c$$

for some computable function  $f$  and some  $c \in \mathbb{N}$ .

Even this relaxed notion of tractability does not seem to be achievable for general structures  $A$ . However, fixed-parameter tractable algorithms have

---

VSAM by removing the explicit reference to indices of records (which are artefacts of the way a database is stored).



been found for model checking on restricted classes of structures, such as those whose Gaifman graphs have bounded treewidth, are planar, exclude certain graphs as minors or topological minors, or which are *sparse* in some precise sense. Results asserting the existence of such algorithms are commonly referred to as *algorithmic meta-theorems*, because they yield efficient algorithms for a wide range of problems. In Part III we investigate in how far algorithmic meta-theorems for first-order and monadic second-order logic can be generalised to order-invariant and successor-invariant counterparts of these logics.

## Organisation of this Thesis

This thesis is roughly divided into three parts:

- In Part I we review notions from logic and graph theory and introduce logics with invariantly used relations.
- Part II deals with the expressive power of logics with invariantly used relations in comparison to their plain counterparts. In cases where the expressive power can be shown to increase, oftentimes the separating queries are made up of structures that are far from being “tame”, leaving open the question of whether there are collapse results on tame structures.

In cases where collapse results are shown it remains to compare the *succinctness* of logics with invariantly used relations to their plain counterparts.

- In Part III we review known algorithmic meta-theorems and investigate in how far they can be extended to logics with invariantly used relations.

We give a comprehensive survey of known results about logics with invariantly used relations. Our own contributions to this field have been published in [EKK13, EEH14, EK16, EEH17, EK17, EvdHK<sup>+</sup>19] and are clearly marked in the corresponding chapters. The presentation of these results has been unified and extended by background material, but in particular parts of the proofs given of our own results have been taken from these (co-authored) papers.

## Acknowledgements

It is my pleasure to give thanks to those who have supported me during the long process of preparing this thesis. The research that led to this thesis has

been conducted while I was a postdoctoral fellow at the National Institute of Informatics in Tokyo, and during my time as a postdoctoral fellow at TU Darmstadt. I am very grateful to my advisors, Ken-ichi Kawarabayashi and Martin Otto, for their support and guidance. Even though Martin was not directly involved with the results presented here, both my way of thinking about finite model theory and my style in presenting the results have been heavily influenced by his example.

Stephan Kreutzer and his group at TU Berlin (in particular Roman Rabinovich and Sebastian Siebertz) have hosted me several times, each of which was both enjoyable and fruitful. The same goes for the joint research with Michael Elberfeld and Frederik Harwath.

Writing this thesis was at times an exhausting process, but I hope reading it will be less painful. I am very grateful to Mikołaj Bojańczyk, Rod Downey, Marc Pfetsch, and Martin Otto for agreeing to referee this thesis.

Last but not least I would like to thank Yumiko, Kyo and Miona for being patient when my mind was (well, is) on my research even though physically I am not at work – and for taking my mind off research, of course.



# Part I

## Background



# Chapter 1

## Preliminaries and Notation

### Commonly Used Notation

We denote by  $\mathbb{N}^+ = \{1, 2, \dots\}$  the set of positive natural numbers and by  $\mathbb{N} = \{0\} \cup \mathbb{N}^+$  the set of natural numbers including 0. For natural numbers  $m \leq n$  we set

$$[n] := \{1, \dots, n\} \quad \text{and} \quad [m, n] := \{m, m+1, \dots, n\}$$

For natural numbers  $a, b \in \mathbb{N}$  and  $k \in \mathbb{N}^+$  we set

$$a \equiv_{\text{mod } k} b \quad :\Leftrightarrow \quad k \text{ divides } (b - a).$$

For a set  $X$  we denote by

$$2^X := \{S \mid S \subseteq X\}$$

its powerset and for  $k \in \mathbb{N}$  by

$$\binom{X}{k} := \{S \subseteq X \mid |S| = k\} \subseteq 2^X$$

the set of  $k$ -element subsets of  $X$ .

We define the *d-fold exponential function*  $d\text{-exp}(n)$  recursively by

$$\begin{aligned} 0\text{-exp}(n) &:= n, \text{ and} \\ (d+1)\text{-exp}(n) &:= 2^{d\text{-exp}(n)}. \end{aligned}$$

The class of functions that grow at most  $d$ -fold exponentially is

$$d\text{-EXP} := \{f : \mathbb{N} \rightarrow \mathbb{N} \mid f(n) \leq d \cdot \exp(n^c) \text{ for some } c \in \mathbb{N} \text{ and all } n > c\}.$$

## 1.1 Logic Preliminaries

Our terminology and notation largely follows that of Ebbinghaus, Flum and Thomas [EFT94, EF99].

### Structures

**Definition 1.1.1** (signatures, structures). A (finite) *signature* is a set

$$\sigma = \{c_1, \dots, c_k, R_1, \dots, R_\ell, f_1, \dots, f_m\}$$

of *constant symbols*  $c_i$ , *relation symbols*  $R_i$ , and *function symbols*  $f_i$ . The *arity* of a relation or function symbol is denoted by  $\text{ar}(R)$  and  $\text{ar}(f)$ , respectively. We will mostly be concerned with relational signatures (i.e.  $m = 0$ ), or with signatures in which all function symbols have arity one.

A  $\sigma$ -*structure*  $A$  is a tuple

$$(V(A), c_1^A, \dots, c_k^A, R_1^A, \dots, R_\ell^A, f_1^A, \dots, f_m^A)$$

consisting of

- a set  $V(A)$  (called the *universe* of  $A$ ),
- an element  $c_i^A \in V(A)$  for each constant symbol  $c_i \in \sigma$ ,
- a relation  $R_i^A \subseteq V(A)^{\text{ar}(R)}$  for each relation symbol  $R_i \in \sigma$ , and
- a function  $f_i^A : V(A)^{\text{ar}(f)} \rightarrow V(A)$  for each function symbol  $f_i \in \sigma$ .

A structure is called *finite* if its universe is finite. The class of all finite  $\sigma$ -structures is denoted by  $\text{Fin}(\sigma)$ .

**Definition 1.1.2** (isomorphism, class, query). Two  $\sigma$ -structures  $A$  and  $B$  are *isomorphic*, written  $A \simeq B$ , if there is a bijection  $\pi : V(A) \rightarrow V(B)$  that is compatible with the interpretations of all symbols from  $\sigma$  in the sense that

- $c^B = \pi(c^A)$  for each constant symbol  $c \in \sigma$ ,
- $(a_1, \dots, a_r) \in R^A$  if, and only if,  $(\pi(a_1), \dots, \pi(a_r)) \in R^B$  for each relation symbol  $R \in \sigma$  of arity  $r = \text{ar}(R)$  and all  $a_1, \dots, a_r \in V(A)$ , and
- $f^B(\pi(a_1), \dots, \pi(a_r)) = \pi(f^A(a_1, \dots, a_r))$  for each function symbol  $f \in \sigma$  of arity  $r = \text{ar}(f)$  and all  $a_1, \dots, a_r \in V(A)$ .

In this case  $\pi$  is called an *isomorphism* between  $A$  and  $B$ . When we speak of a *class*  $\mathcal{C}$  of structures, we mean a collection of structures that is closed under isomorphisms, i.e. if  $A \in \mathcal{C}$  and  $B \simeq A$ , then  $B \in \mathcal{C}$ .

We will mostly be concerned with classes containing only finite structures and call these *queries*. In particular, we say that a query is definable in a logic  $\mathbf{L}$  if it is the class of *finite* models of some sentence in  $\mathbf{L}$ .

**Definition 1.1.3** (expansions, reducts). Let  $\sigma$  and  $\tau$  be two signatures such that  $\sigma \subseteq \tau$ , let  $A$  be a  $\sigma$ -structure and  $B$  a  $\tau$ -structure such that

- $V(A) = V(B)$ ,
- $c^A = c^B$  for all constant symbols  $c \in \sigma$ ,
- $R^A = R^B \cap V(A)^{\text{ar}(R)}$  for all relation symbols  $R \in \sigma$ , and
- $f^A(a_1, \dots, a_r) = f^B(a_1, \dots, a_r) \in V(A)$  for all function symbols  $f \in \sigma$  of arity  $r = \text{ar}(f)$  and  $a_1, \dots, a_r \in V(A)$ .

In other words,  $A$  and  $B$  have the same universe and agree on all constants, relations, and functions in  $\sigma$ . Then  $B$  is called a  $\tau$ -*expansion* of  $A$ , and  $A$  is called the  $\sigma$ -*reduct* of  $B$ , denoted by  $A = B|_\sigma$ .

**Definition 1.1.4** ((disjoint) unions). Let  $\sigma$  be a relational signature without constant symbols and  $A, B$  two  $\sigma$ -structures. Their *union* is the structure  $C := A \cup B$  with universe  $V(C) := V(A) \cup V(B)$  and relations interpreted by  $R^C := R^A \cup R^B$ .

If the universes  $V(A)$  and  $V(B)$  are disjoint we write the union as  $A \sqcup B$  for emphasis.

In Parts II and III we will be concerned with the behaviour of certain logics on restricted classes of structures. These restrictions will be in terms of the following graph:

**Definition 1.1.5** (Gaifman graph). Let  $\sigma$  be a signature and  $A$  a  $\sigma$ -structure. The *Gaifman graph*  $G_A$  of  $A$  is the graph with vertex set  $V(A)$  and an edge  $uv$  between any two distinct vertices  $u, v \in V(A)$  for which

- there is a relation symbol  $R \in \sigma$  and a tuple  $(a_1, \dots, a_r) \in R(A)$  for which  $u, v \in \{a_1, \dots, a_r\}$ , or
- there is an  $r$ -ary function symbol  $f \in \sigma$  and elements  $a_1, \dots, a_r \in V(A)$  for which  $u, v \in \{a_1, \dots, a_r, f(a_1, \dots, a_r)\}$ .

Note that the Gaifman graph degenerates into a clique as soon as there is a function symbol  $f \in \sigma$  of arity at least two.



## First-Order Logic

We fix a countably infinite set  $\mathcal{V} = \{v_0, v_1, \dots\}$  whose elements we call first-order variables. Elements of  $\mathcal{V}$  will often be denoted by  $x, y, x_1, x_2$ , etc., and they may be bound to elements of the universe of a structure. Formally, terms and formulae of first-order logic are strings over the alphabet

$$\mathcal{A}_\sigma := \mathcal{V} \cup \sigma \cup \{\forall, \exists, \doteq, \wedge, \vee, \neg, (, )\},$$

and we tacitly assume that the three sets in the above union are disjoint. We write function applications in prefix-order and therefore do not need to enclose parameters in brackets or separate them in any way; as strings over the alphabet  $\mathcal{A}_\sigma^*$ , terms and finite sequences of terms are uniquely parsable.

**Definition 1.1.6** (terms). For a given signature  $\sigma$ , the set  $T(\sigma) \subseteq \mathcal{A}_\sigma^*$  of  $\sigma$ -terms is the smallest set for which

- $\mathcal{V} \subseteq T(\sigma)$ ,
- $c \in T(\sigma)$  for every constant symbol  $c \in \sigma$ , and
- $ft_1 \dots t_r \in T(\sigma)$  whenever  $f \in \sigma$  is a function symbol of arity  $r$  and  $t_1, \dots, t_r \in T(\sigma)$ .

The set  $\text{var}(t)$  of variables occurring in a term  $t \in T(\sigma)$  is defined inductively by

$$\begin{aligned} \text{var}(v_i) &:= \{v_i\} \text{ for every variable } v_i \in \mathcal{V}, \\ \text{var}(c) &:= \emptyset \text{ for every constant symbol } c \in \sigma, \text{ and} \\ \text{var}(ft_1 \dots t_r) &:= \bigcup_{i=1}^r \text{var}(t_i) \text{ otherwise.} \end{aligned}$$

**Definition 1.1.7** (variable binding, interpretation). Let  $\sigma$  be a signature and  $A$  a  $\sigma$ -structure. A *variable binding* in  $A$  is a function  $\beta : \mathcal{V} \rightarrow V(A)$ . A pair  $\mathfrak{I} = (A, \beta)$  consisting of a  $\sigma$ -structure  $A$  and a variable binding  $\beta$  in  $A$  is called an *interpretation*. For a variable binding  $\beta$ , a variable  $x \in \mathcal{V}$  and  $a \in V(A)$  we define the variable binding  $\beta_x^a$  by

$$\left(\beta_x^a\right)(y) := \begin{cases} a & \text{if } x = y, \\ \beta(y) & \text{otherwise.} \end{cases}$$

If  $\mathfrak{I} = (A, \beta)$  we write  $\mathfrak{I}_x^a$  for the interpretation  $(A, \beta_x^a)$ .

For each interpretation  $\mathfrak{I} = (A, \beta)$  we inductively define a function

$$[\cdot]_{\mathfrak{I}} : T(\sigma) \rightarrow V(A)$$

by

- $[v_i]_{\mathfrak{I}} := \beta(v_i)$  for each variable  $v_i \in \mathcal{V}$ ,
- $[c]_{\mathfrak{I}} := c^A$  for each constant symbol  $c \in \sigma$ , and
- $[ft_1 \dots t_r]_{\mathfrak{I}} := f^A([t_1]_{\mathfrak{I}}, \dots, [t_r]_{\mathfrak{I}})$  for each function symbol  $f \in \sigma$  of arity  $r$ .

With these preparations we can define first-order logic:

**Definition 1.1.8** (first-order logic). The set  $\mathbf{FO}(\sigma) \subseteq \mathcal{A}_{\sigma}^*$  of *first-order logical formulae over  $\sigma$*  is defined as the smallest set that contains

- formulae  $\top$  and  $\perp$  (for “true” and “false”),
- formulae  $t_1 \doteq t_2$  for all terms  $t_1, t_2 \in T(\sigma)$ ,
- formulae  $Rt_1 \dots t_r$  for all  $R \in \sigma$  of arity  $r$  and terms  $t_1, \dots, t_r \in T(\sigma)$ ,
- a formula  $\neg\varphi$  for every  $\varphi \in \mathbf{FO}(\sigma)$ ,
- formulae  $(\varphi_1 \vee \varphi_2)$  and  $(\varphi_1 \wedge \varphi_2)$  for every  $\varphi_1, \varphi_2 \in \mathbf{FO}(\sigma)$ ,
- formulae  $\exists x \varphi$  and  $\forall x \varphi$  for every  $x \in \mathcal{V}$  and  $\varphi \in \mathbf{FO}(\sigma)$ .

For easier readability we omit parentheses when writing FO-formulae if no confusion seems likely. Formulae of the form  $t_1 \doteq t_2$  and  $Rt_1 \dots t_r$  are called *atomic formulae*.

**Definition 1.1.9** (free variables). For a formula  $\varphi \in \mathbf{FO}$ , the set  $\text{free}(\varphi) \subseteq \mathcal{V}$  of *free variables* is defined inductively by

$$\begin{aligned} \text{free}(\top), \text{free}(\perp) &:= \emptyset \\ \text{free}(t_1 \doteq t_2) &:= \text{var}(t_1) \cup \text{var}(t_2), \\ \text{free}(Rt_1 \dots t_r) &:= \bigcup_{i=1}^r \text{var}(t_i) \\ \text{free}(\neg\varphi) &:= \text{free}(\varphi) \\ \text{free}(\varphi_1 \wedge \varphi_2) &:= \text{free}(\varphi_1) \cup \text{free}(\varphi_2) \\ \text{free}(\varphi_1 \vee \varphi_2) &:= \text{free}(\varphi_1) \cup \text{free}(\varphi_2) \\ \text{free}(\exists x \varphi) &:= \text{free}(\varphi) \setminus \{x\} \\ \text{free}(\forall x \varphi) &:= \text{free}(\varphi) \setminus \{x\} \end{aligned}$$

A formula  $\varphi \in \mathbf{FO}$  with  $\text{free}(\varphi) = \emptyset$  is called a *sentence*. If  $\text{free}(\varphi) \subseteq \{v_1, \dots, v_n\}$  we sometimes write  $\varphi(\bar{x})$  or  $\varphi(x_1, \dots, x_n)$  to emphasize this fact

**Definition 1.1.10** (semantics of FO). We define a relation  $\models$  between interpretations  $\mathfrak{I} = (A, \beta)$  and formulae  $\varphi \in \text{FO}(\sigma)$  as follows:

- $\mathfrak{I} \models \top$  and  $\mathfrak{I} \not\models \perp$  for every  $\mathfrak{I}$
- $\mathfrak{I} \models t_1 \doteq t_2$  if  $[t_1]_{\mathfrak{I}} = [t_2]_{\mathfrak{I}}$ ,
- $\mathfrak{I} \models Rt_1 \dots t_r$  if  $([t_1]_{\mathfrak{I}}, \dots, [t_r]_{\mathfrak{I}}) \in R^A$ ,
- $\mathfrak{I} \models \neg\varphi$  if not  $\mathfrak{I} \models \varphi$ ,
- $\mathfrak{I} \models \varphi_1 \vee \varphi_2$  if  $\mathfrak{I} \models \varphi_1$  or  $\mathfrak{I} \models \varphi_2$ ,
- $\mathfrak{I} \models \varphi_1 \wedge \varphi_2$  if  $\mathfrak{I} \models \varphi_1$  and  $\mathfrak{I} \models \varphi_2$ ,
- $\mathfrak{I} \models \exists x \varphi$  if there exists an  $a \in V(A)$  such that  $\mathfrak{I}_x^a \models \varphi$ , and
- $\mathfrak{I} \models \forall x \varphi$  if  $\mathfrak{I}_x^a \models \varphi$  for all  $a \in V(A)$ .

If  $\mathfrak{I} \models \varphi$  we say that  $\varphi$  is *satisfied* in  $\mathfrak{I}$ , or that  $\mathfrak{I}$  is a *model* of  $\mathfrak{I}$ .

It is straight-forward to show that whether  $\mathfrak{I} \models \varphi$  for a formula  $\varphi$  and an interpretation  $\mathfrak{I}$  is independent of which values are bound to variables that are not free in  $\varphi$ : If  $A$  is a structure and  $\beta, \beta' : \mathcal{V} \rightarrow V(A)$  satisfy  $\beta(x) = \beta'(x)$  for all  $x \in \text{free}(\varphi)$ , then

$$(A, \beta) \models \varphi \quad \text{if, and only if,} \quad (A, \beta') \models \varphi.$$

In particular, if  $\varphi$  is a sentence we just write  $A \models \varphi$  instead of  $(A, \beta) \models \varphi$  and say that  $A$  is a model of  $\varphi$  in this case. A class  $\mathcal{C}$  of  $\sigma$ -structures is called *elementary* if it is the class of all models of some sentence  $\varphi \in \text{FO}(\sigma)$ .

For a tuple  $\bar{a} = (a_1, \dots, a_n) \in V(A)^n$  write

$$A, \bar{a} \models \varphi \quad \text{or} \quad A \models \varphi[\bar{a}]$$

if  $(A, \beta) \models \varphi$  for some (or, equivalently, any)  $\beta$  which has  $\beta(v_i) = a_i$  for  $i \in [n]$ .

For a class  $\mathcal{C}$  of  $\sigma$ -structures and formulae  $\varphi$  and  $\psi$  we say that  $\varphi$  is *equivalent to  $\psi$  over  $\mathcal{C}$*  if for all interpretations  $\mathfrak{I} = (A, \beta)$  with structures  $A \in \mathcal{C}$  we have

$$\mathfrak{I} \models \varphi \quad \text{iff} \quad \mathfrak{I} \models \psi.$$

We write  $\varphi \equiv^{\mathcal{C}} \psi$  in this case, and if  $\mathcal{C} = \mathbf{Fin}$  is the class of all finite structures we just write  $\varphi \equiv \psi$ . For example:

- $(x \doteq y) \equiv (y \doteq x)$
- $Exy \equiv^{\mathbf{Graph}} Eyx$ , but not  $Exy \equiv Eyx$
- $\exists x \varphi \equiv \neg \forall x \neg \varphi$  for all signatures  $\sigma$  and all  $\text{FO}(\sigma)$ -formulae  $\varphi$

**Definition 1.1.11.** The *quantifier rank*  $\text{qr}(\varphi)$  of a formula  $\varphi$  is defined recursively as

$$\begin{aligned} \text{qr}(\top), \text{qr}(\perp) &:= 0, \\ \text{qr}(x \dot{=} y) &:= 0, \\ \text{qr}(Rx_1 \dots x_r) &:= 0, \\ \text{qr}(\neg\varphi) &:= \text{qr}(\varphi), \\ \text{qr}(\varphi_1 \wedge \varphi_2), \text{qr}(\varphi_1 \vee \varphi_2) &:= \max\{\text{qr}(\varphi_1), \text{qr}(\varphi_2)\}, \\ \text{qr}(\exists x \varphi), \text{qr}(\forall x \varphi) &:= 1 + \text{qr}(\varphi). \end{aligned}$$

For  $q \geq 0$  we set

$$\text{FO}_q(\sigma) := \{\varphi \in \text{FO}(\sigma) \mid \text{qr}(\varphi) \leq q\}.$$

A formula  $\varphi$  is in *negation normal form* if negation symbols appear only in front of atomic formulae. For a given formula one can efficiently compute an equivalent formula in negation normal form.

The *quantifier alternation depth*  $\text{qad}(\varphi)$  of a formula  $\varphi$  in negation normal form is the maximum number of quantifier alternations on a path in the syntax tree of  $\varphi$ . For  $\varphi$  not in negation normal form we first compute an equivalent formula  $\varphi'$  in negation normal form (using some fixed algorithm) and then set  $\text{qad}(\varphi) := \text{qad}(\varphi')$ .

## FO with Modulo Counting

We will need the extension of first-order logic with counting quantifiers:

**Definition 1.1.12 (FMod).** We define formulae of *first-order logic with modulo counting quantifiers* (FMod) inductively using the same rules as those for FO. Additionally, for every number  $p \in \mathbb{N}^+$  and  $k \in \mathbb{N}$ ,

$$\exists^{k \pmod p} x \varphi$$

is a formula of FMod whenever  $\varphi \in \text{FMod}$ . For an interpretation  $\mathfrak{I} = (A, \beta)$  and  $\varphi \in \text{FMod}(\sigma)$  we define  $\mathfrak{I} \models \varphi$  recursively as for FO, setting

$$\mathfrak{I} \models \exists^{k \pmod p} x \varphi \quad :\Leftrightarrow \quad \left| \left\{ a \in V(A) \mid \mathfrak{I} \frac{a}{x} \models \varphi \right\} \right| \equiv_{\text{mod } p} k.$$

The logic **FMod** has strictly more expressive power than **FO**, as for example the class of all finite sets of even cardinality is definable in **FMod** but not in **FO**.

## Monadic Second-Order Logic

While first-order logic only allows quantification over elements of  $V(A)$ , second-order logic allows quantification over arbitrary relations  $R \subseteq V(A)^r$ . This is a very powerful logic, as witnessed for example by the fact that existential second-order logic captures the complexity class NP of non-deterministic polynomial time (cf. [EF99]). Restricting the arity of quantified relations to 1 (i.e. allowing only quantification over sets) yields a logic which strikes a good balance between expressiveness and feasibility on many classes of finite structures.

For this we fix a set  $\mathcal{V}_{\text{set}} = \{V_0, V_1, \dots\}$  whose elements we call set variables. We will often denote them using other letters such as  $X, Y, \dots$ , but to distinguish them from first-order variables we will always denote them by capital letters. We extend the alphabet  $A_\sigma$  of first-order logic to  $A_\sigma \cup \mathcal{V}_{\text{set}}$ .

**Definition 1.1.13** (monadic second-order logic). Terms of monadic second-order logic (**MSO**) are exactly those of **FO**. Formulae of **MSO** are defined inductively using the rules for formulae of **FO**, where additionally

- $Xt \in \text{MSO}(\sigma)$  for every term  $t \in T(\sigma)$  and  $X \in \mathcal{V}_{\text{set}}$ , and
- $\exists X \varphi \in \text{MSO}(\sigma)$  and  $\forall X \varphi \in \text{MSO}(\sigma)$  whenever  $X \in \mathcal{V}_{\text{set}}$  and  $\varphi \in \text{MSO}(\sigma)$ .

For a  $\sigma$ -structure  $A$ , an **MSO**-variable binding in  $A$  is a pair of functions  $\beta : \mathcal{V} \rightarrow V(A)$  and  $\beta_{\text{set}} : \mathcal{V}_{\text{set}} \rightarrow \mathcal{P}(V(A))$ ; accordingly, an interpretation is a triple  $\mathfrak{J} = (A, \beta, \beta_{\text{set}})$ . For an interpretation  $\mathfrak{J} = (A, \beta, \beta_{\text{set}})$  we set

$$\begin{aligned} \mathfrak{J} \models Xt & \quad :\Leftrightarrow \quad [t]_{\mathfrak{J}} \in \beta_{\text{set}}(X), \\ \mathfrak{J} \models \exists X \varphi & \quad :\Leftrightarrow \quad \mathfrak{J} \frac{M}{X} \models \varphi \text{ for some } M \subseteq V(A), \text{ and} \\ \mathfrak{J} \models \forall X \varphi & \quad :\Leftrightarrow \quad \mathfrak{J} \frac{M}{X} \models \varphi \text{ for all } M \subseteq V(A). \end{aligned}$$

We extend the definition of  $\text{free}(\varphi)$  and  $\text{qr}(\varphi)$  to  $\varphi \in \text{MSO}$  using

$$\begin{aligned}\text{free}(Xt) &:= \{X\} \cup \text{var}(t), \\ \text{free}(\exists X \varphi) = \text{free}(\forall X \varphi) &:= \text{free}(\varphi) \setminus \{X\}, \\ \text{qr}(Xt) &:= 0, \text{ and} \\ \text{qr}(\exists X \varphi) = \text{qr}(\forall X \varphi) &:= \text{qr}(\varphi) + 1.\end{aligned}$$

Again we refer to the set of **MSO** formulae of quantifier rank at most  $q$  by  $\text{MSO}_q$ .

For the case of graphs, our definition of **MSO** yields a logic which is sometimes referred to as  $\text{MSO}_1$ , as opposed to the logic  $\text{MSO}_2$  which allows quantification not only over vertices and sets of vertices, but also over sets of edges, cf. [Cou03]; note that in this notation the subscript does *not* specify a bound on the quantifier rank. For arbitrary relational signatures one can allow second-order quantification over subsets of existing relations, obtaining a logic called *guarded second-order logic*, cf. [Blu10]. While these logics are in general stronger than monadic second-order logic in our sense, their expressive powers coincide on important classes of structures such as those of bounded treewidth (cf. Section 3.1 and [Cou03]).

## MSO with Modulo Counting

Similar to **FMod** we define an extension of monadic second-order logic by the ability to count modulo some fixed number. Since **MSO** allows quantification over sets, we can introduce this kind of counting using atomic formulae, instead of special quantifiers as in the case of **FMod**. We call the resulting logic *counting MSO* (**CMSO**) rather than  $\text{MSO}_{\text{mod}}$ , as it has come to be called this way in the literature.

**Definition 1.1.14.** Formulae of *counting MSO* (**CMSO**) are formed like those of **MSO** with additional atomic formulae of the form

$$\#_n X,$$

where  $n \geq 2$  and  $X \in \mathcal{V}_{\text{set}}$  is a set variable. For an interpretation  $\mathfrak{J} = (A, \beta, \beta_{\text{set}})$  we set

$$\mathfrak{J} \models \#_n X \quad \text{if, and only if,} \quad n \text{ divides } |\beta_{\text{set}}(X)|.$$

Note that we could have allowed formulae of the form  $\#_n^k X$  with

the intended meaning that  $|X| \equiv_{\text{mod } n} k$  for  $n \geq 2$  and  $0 \leq k < n$ , but these can be rephrased in our setting (at the cost of introducing additional existential quantifiers).

## Types and Composition Theorems

Let  $L \in \{\text{FO}, \text{MSO}, \text{FOMod}, \text{CMSO}\}$ ,  $q \in \mathbb{N}$  and let  $\sigma$  be a relational signature. For two structures  $A$  and  $B$  and tuples  $\bar{a} \in V(A)^k$  and  $\bar{b} \in V(B)^k$  of equal length  $k$  we say that  $(A, \bar{a})$  and  $(B, \bar{b})$  are  $L_q$ -equivalent, written  $A, \bar{a} \equiv_q^L B, \bar{b}$ , if

$$A \models \varphi[\bar{a}] \quad \text{iff} \quad B \models \varphi[\bar{b}]$$

for all  $\varphi(x_1, \dots, x_k) \in L_q(\sigma)$ . The  $L_q$ -type of  $\bar{a}$  in  $A$  is the set

$$\text{tp}_q^L(A, \bar{a}) := \{\varphi(x_1, \dots, x_k) \in L_q(\sigma) \mid A \models \varphi[\bar{a}]\},$$

In particular, if  $\bar{a}$  and  $\bar{b}$  are the empty tuples, we get the  $q$ -type of  $A$ . When the logic  $L$  we refer to is clear from the context we omit the index  $L$  and speak of the  $q$ -type  $\text{tp}_q(A, \bar{a})$  and  $q$ -equivalence  $\equiv_q$ .

The notions of types and equivalence are related by the following obvious lemma:

**Lemma 1.1.15.** *For  $L \in \{\text{FO}, \text{MSO}\}$  and  $q \in \mathbb{N}$ ,*

$$A, \bar{a} \equiv_q^L B, \bar{b} \quad \text{if, and only if,} \quad \text{tp}_q^L(A, \bar{a}) = \text{tp}_q^L(B, \bar{b}).$$

*Proof.* Immediate from the definitions. □

It is well known that the type of the disjoint union of two structures is determined by the types of their components, a result commonly referred to as *Feferman-Vaught Theorem*, cf. [Mak04]. This can be extended to other settings, such as non-disjoint unions or ordered unions, as well as for other logics than FO and MSO. We will need the following instantiations:

**Theorem 1.1.16** (Feferman-Vaught Theorem). *Let  $L \in \{\text{FO}, \text{MSO}\}$ ,  $q \in \mathbb{N}$  and let  $\sigma$  be a relational signature.*

1. *ordered disjoint sum: Let  $(A_1, \preceq^{A_1}), (A_2, \preceq^{A_2}), (B_1, \preceq^{B_1}), (B_2, \preceq^{B_2})$  be ordered  $\sigma$ -structures. If*

$$(A_1, \preceq^{A_1}) \equiv_q^L (A_2, \preceq^{A_2}) \quad \text{and} \quad (B_1, \preceq^{B_1}) \equiv_q^L (B_2, \preceq^{B_2}),$$

then

$$(A_1 \sqcup B_1, \preceq^{A_1} + \preceq^{B_1}) \equiv_q^L (A_2 \sqcup B_2, \preceq^{A_2} \oplus \preceq^{B_2}).$$

In other words,  $\text{tp}_q^L(A_1 \sqcup B_1, \preceq^{A_1} \oplus \preceq^{B_1})$  is determined by  $\text{tp}_q^L(A_1, \preceq^{A_1})$  and  $\text{tp}_q^L(B_1, \preceq^{B_1})$ .

Here,  $\preceq_1 \oplus \preceq_2$  denotes the ordered sum of two linear orders  $\preceq_1 \subseteq M_1^2$  and  $\preceq_2 \subseteq M_2^2$  on disjoint sets  $M_1, M_2$ :

$$\preceq_1 \oplus \preceq_2 := \preceq_1 \cup \preceq_2 \cup \{(a, b) \mid a \in M_1, b \in M_2\}$$

2. (possibly) non-disjoint sum: Let  $A$  and  $B$  be two  $\sigma$ -structures, and let  $\bar{c} = (c_1, \dots, c_k)$  be a tuple such that  $\{c_1, \dots, c_k\} = V(A) \cap V(B)$ . Then  $\text{tp}_q^L(\bar{c}, A \cup B)$  is determined by  $\text{tp}_q^L(\bar{c}, A)$  and  $\text{tp}_q^L(\bar{c}, B)$ .

Furthermore, the types of the union structures can be computed from the types of their component structures.

We refer to Makowsky's paper [Mak04] for a proof.





## Chapter 2

# Logics with Invariantly Used Relations

### 2.1 Invariantly Used Relations

In this chapter we will be dealing with logics that may use relation symbols that are not defined on the structures they talk about. Instead, when evaluating the truth value of a sentence in such a logic in a given structure, we expand the structure by interpreting these additional relation symbols so that they form, for example, a linear order or its successor relation on the elements of the structure. We arbitrarily choose from among all suitable relations and restrict attention to those sentences whose truth value on all finite structures is independent of the particular choice.

In its most general form this leads to the following definition:

**Definition 2.1.1** ( $\mathcal{R}$ -invariance). Let  $\tau$  be a relational signature and  $\mathcal{R}$  a class of  $\tau$ -structures such that for each  $n \geq 1$  there is at least one  $A_n \in \mathcal{R}$  with  $|V(A_n)| = n$ . Let  $\sigma$  be another signature with  $\sigma \cap \tau = \emptyset$ , and  $A$  a finite  $\sigma$ -structure. A sentence  $\varphi$  in some logic  $L(\sigma \cup \tau)$  is called  *$\mathcal{R}$ -invariant on  $A$*  if for any two  $(\sigma \cup \tau)$ -expansions  $B_1, B_2$  of  $A$  with  $B_1|_\tau, B_2|_\tau \in \mathcal{R}$  we have

$$B_1 \models \varphi \text{ if, and only if, } B_2 \models \varphi.$$

In this case we say that  $A \models \varphi$  if  $B \models \varphi$  for some (or, equivalently, for every) such expansion. In cases where  $\tau = \{R\}$  contains just a single relation symbol we sometimes denote the satisfaction relation by  $\models_R$  rather than just  $\models$  to stress the fact that the relation  $R$  is used invariantly. For a class  $\mathcal{C}$  of finite  $\sigma$ -structures we call  $\varphi$   *$\mathcal{R}$ -invariant on  $\mathcal{C}$*  if it is  $\mathcal{R}$ -invariant on all

$A \in \mathcal{C}$ .

The set of all  $\mathbf{L}$ -sentences that are  $\mathcal{R}$ -invariant on  $\mathbf{Fin}$  is denoted by  $\mathcal{R}\text{-inv-L}$ .

While it makes sense to speak of  $\mathcal{R}$ -invariance also for infinite structures, the following well-known result from model theory shows that, at least for elementary classes  $\mathcal{R}$ , this does not increase the expressive power of first-order logic:

**Theorem 2.1.2.** *Let  $\tau$  be a relational signature and  $\mathcal{R}$  an elementary class of  $\tau$ -structures, say  $(A \in \mathcal{R} \Leftrightarrow A \models \rho)$  for some  $\rho \in \mathbf{FO}(\tau)$ . Then if  $\varphi \in \mathbf{FO}(\sigma \cup \tau)$  is  $\mathcal{R}$ -invariant on all structures, there is an equivalent  $\psi \in \mathbf{FO}(\sigma)$ , i.e. such that*

$$B \models \varphi \text{ if, and only if, } A \models \psi \quad (2.1)$$

for all  $\sigma$ -structures  $A$  and  $(\sigma \cup \tau)$ -structures  $B$  with  $B|_{\tau} \in \mathcal{R}$  and  $B|_{\sigma} \simeq A$ .

*Proof.* Let  $\tau'$  be a disjoint copy of  $\tau$ , disjoint from  $\sigma$ , and let  $\rho' \in \mathbf{FO}(\tau')$  be obtained from  $\rho$  by replacing the relation symbols from  $\tau$  by their counterparts in  $\tau'$ , and similarly for  $\varphi'$ . Then

$$\rho \wedge \varphi \models \rho' \rightarrow \varphi',$$

and by Craig's Interpolation Theorem (cf. [CK90, Thm. 2.2.20]) there is an interpolant  $\psi \in \mathbf{FO}(\sigma)$ , i.e. a formula such that

$$\rho \wedge \varphi \models \psi \models \rho' \rightarrow \varphi.$$

This  $\psi$  satisfies (2.1). □

We now introduce the logics will mainly be concerned with, namely order-invariant and successor-invariant variants of first-order and monadic second-order logic.

## Order-Invariant Logics

Let  $\mathcal{R}_{<}$  be the class of finite linear orders, i.e.  $\{<\}$ -structures isomorphic to some structure  $A$  with universe  $V(A) = [n]$  and

$$<^A = \{(i, j) \mid 1 \leq i < j \leq n\}.$$

Note that this class is the class of finite models of the elementary class of all linearly ordered structures:

$$A \in \mathcal{R}_{<} \quad \text{if, and only if,} \quad A \text{ is finite and } A \models \varphi_{<},$$

where

$$\begin{aligned} \varphi_{<} := & \forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z) \wedge \\ & \forall x \forall y (x < y \vee x \dot{=} y \vee y < x) \wedge \forall x \forall y \neg (x < y \wedge y < x) \end{aligned}$$

Simplifying notation, we denote  $\mathcal{R}_{<}$ -invariant FO by  $<\text{-inv-FO}$ , and similarly for other logics. Furthermore, we define  $\sigma^{<} := \sigma \cup \{<\}$  provided  $\sigma$  is a finite relational signature not containing  $<$ . For a  $\sigma$ -structure  $A$ , an *ordered expansion* is a  $\sigma^{<}$ -structure  $A'$  such that

$$A'|_{\sigma} = A \quad \text{and} \quad A'|_{\{<\}} \in \mathcal{R}_{<}.$$

Order-invariant logics appear naturally in various branches of finite model theory, cf. [Sch13] for a survey. We present a few examples:

**Order-invariant fixed-point logics** In descriptive complexity theory, it is well known that least fixed-point logic captures polynomial time on ordered structures: A class  $\mathcal{C}$  of finite ordered structures is axiomatisable in LFP if, and only if, it is decidable in polynomial time ([Imm82, Var82, Liv82], cf. [EF99, Ch. 7]).

The proof yields an order-invariant LFP-sentence for each polynomial-time decidable class of structures, so the theorem can be restated as:

$<\text{-inv-LFP}$  captures polynomial time on the class of all finite structures.

This result has the serious drawback that the syntax of  $<\text{-inv-LFP}$  is undecidable (as we shall see below), i.e. given a sentence  $\varphi \in \text{LFP}(\sigma \cup \{<\})$ , we can not algorithmically decide whether it is order-invariant or not. The question of whether there is a logic with decidable syntax capturing polynomial time on *all* finite structures remains open and has been a subject of intense research (cf. [Gro08] for a survey).

**Logics with consistent choice** Hilbert introduced a consistent choice operator (called  $\epsilon$ -operator) to first-order logic by allowing terms of the

form

$$(\epsilon_x \varphi(x, \bar{y})),$$

where  $\varphi(x, \bar{y})$  is an FO-formula (possibly using  $\epsilon$ -terms) with free variables among  $x, \bar{y}$ . Given an interpretation  $\mathfrak{I} = (A, \beta)$ , such a formula  $\varphi$  defines a set

$$\varphi(\mathfrak{I}) := \{a \in V(A) \mid \mathfrak{I} \frac{a}{x} \models \varphi\},$$

and the  $\epsilon_x$ -operator consistently chooses one element of this set, i.e.

$$[\epsilon_x \varphi(x, \bar{y})]_{\mathfrak{I}} \in \varphi(\mathfrak{I})$$

and  $[\epsilon_x \varphi(x, \bar{y})]_{\mathfrak{I}}$  is determined by the set  $\varphi(\mathfrak{I})$ . We call the resulting logic  $\epsilon$ -FO.

In the presence of a linear order  $<$  we may choose the  $<$ -minimal element of  $\varphi(\mathfrak{I})$  as  $[\epsilon_x \varphi(x, \bar{y})]_{\mathfrak{I}}$ , showing that  $\epsilon$ -FO is contained in  $<$ -inv-FO. In fact, since  $\epsilon$ -FO is stronger than FO on finite structures (cf. [Ott00]), this implies Gurevich's result that also  $<$ -inv-FO is stronger than FO on finite structures (cf. Section 4.1).

**Model checking** By *model checking* for a given logic  $L$  we mean the following algorithmic problem: Given a finite structure  $A$  and a sentence  $\varphi$  in some logic, check algorithmically whether  $A \models \varphi$ . In this context, the structure  $A$  has to be represented in computer memory, and this is usually done by identifying the set  $V(A)$  with a set  $\{0, 1, \dots, |V(A)| - 1\}$  of natural numbers.

Since elements of  $A$  can now be seen as natural numbers, it makes sense to allow  $\varphi$  access to this linear order. Since the order is an artefact of the chosen encoding, it is natural to restrict attention to those  $\varphi$  that are order-invariant.

## Successor-Invariant FO

Let  $\mathcal{R}_{\text{succ}}$  be the class of successor relations, i.e.  $\{\text{succ}\}$ -structures isomorphic to some structure  $A$  with universe  $V(A) = [n]$  and

$$\text{succ}^A = \{(i, i+1) \mid 1 \leq i < n\}.$$

We denote  $\mathcal{R}_{\text{succ}}$ -invariant FO by  $\text{succ-inv-FO}$ . Again we define  $\sigma^{\text{succ}} := \sigma \cup \{\text{succ}\}$  provided  $\sigma$  is a finite relational signature not containing  $\text{succ}$ . For a  $\sigma$ -structure  $A$ , a *successor expansion* is a  $\sigma^{\text{succ}}$ -structure  $A'$  such that

$$A'|_{\sigma} = A \quad \text{and} \quad A'|_{\{\text{succ}\}} \in \mathcal{R}_{\text{succ}}.$$

Note that unlike  $\mathcal{R}_{<}$ , the class of finite successor-structures is *not* the finite fragment of any elementary class.

Again we can define successor-invariant counterparts of various other logics. Often these coincide with order-invariant logics, though, because in many cases we can recover the natural linear order induced by the successor relation  $\text{succ}$ . This is true for MSO using the MSO-formula

$$\varphi_{<}(x, y) := \forall X \left( \left( Xx \wedge \forall u \forall v ((Xu \wedge \text{succ}uv) \rightarrow Xv) \right) \rightarrow Xy \right),$$

and similarly we obtain  $\text{succ-inv-LFP} = <\text{-inv-LFP}$ .

## Further Arithmetic Relations

Once we identify the elements of the universe  $V(A)$  of a structure with an initial segment of the natural numbers it is natural to add further arithmetic relations such as addition and multiplication. The relative expressive power of these logics has been studied in particular for FO.

**$+ \text{-inv-FO}$**  Even on sets (i.e. structures without any relations defined on them)  $+ \text{-inv-FO}$  is easily seen to be stronger than FO because it can express evenness (by saying that the index of the last element is an odd number, if one starts counting at zero), which is not even possible in MSO. The expressive power of  $+ \text{-inv-FO}$  on sets equals the expressive power of FO on initial segments of  $\mathbb{N}$  with addition, which has been studied by Lynch [Lyn82] and equals that of FO with null-ary predicates for size of the universe modulo some number.

On word structures, i.e. structures with a linear order and an arbitrary number of unary predicates, the expressive power of  $+ \text{-inv-FO}$  has been studied by Schweikardt and Segoufin [SS10]: It remains open whether  $+ \text{-inv-FO}$  can define non-regular languages, but any regular language definable in  $+ \text{-inv-FO}$  can also be defined by FO with null-ary predicates for the size of the universe modulo some number. In particular, this extends Lynch's result from sets to coloured sets.

**$(+, \times)$ -inv-FO** The logic  $(+, \times)$ -inv-FO is known to have considerably more expressive power than both FO and  $+-$ inv-FO. One key insight here is that for a structure  $A$  and a set  $M \subseteq V(A)$  of size polylogarithmic in the size of  $A$ , there is an FO-definable one-to-one mapping between  $M$  and the initial segment  $\{0, \dots, |M| - 1\}$  of  $\mathbb{N}$  (cf. [DLM07] for a proof). In particular,  $(+, \times)$ -inv-FO can count up to a polylogarithmic threshold.

Furthermore, let **Bit** be the binary bit-predicate on natural numbers, i.e. if

$$n = \sum_{i \geq 0} b_i^{(n)} \cdot 2^i$$

with  $b_i^{(n)} \in \{0, 1\}$  for all  $i$  is the binary representation of  $n$ , then

$$(n, i) \in \text{Bit} \quad :\Leftrightarrow \quad b_i^{(n)} = 1.$$

In first-order logic one can define **Bit** from  $+$  and  $\times$  (cf. [Imm99, Sec. 1.2.1]), which gives  $(+, \times)$ -inv-FO the expressive power of MSO on subsets of polylogarithmic size.

**Arb-inv-FO** Finally, we may extend first-order logic by allowing it to invariantly refer to *arbitrary* relations, resulting in the logic **Arb-inv-FO**. Since we allow non-decidable and even non-recursively enumerable relations to be used invariantly, the resulting logic is only interesting in so far as we may prove negative results about it. This is indeed possible, relying on the fact that formulae of **Arb-inv-FO** can be translated into bounded-depth polynomial size families of circuits, one of the few computational models for which unconditional lower bounds have been proven (cf. [Hå86]). Since these lower bounds work for non-uniform families of circuits, they can be used to prove locality results for **Arb-inv-FO**, cf. Section 4.3.

## 2.2 Undecidability of the Syntax

Note that invariant logics are defined in a semantic way, by restricting the set of sentences of a particular logic to those which are invariant under reinterpreting certain relation symbols. This has several unfortunate consequences:

- Even in very simple cases (such as order-invariance on star forests, see below) it is not decidable whether a given L-sentence is invariant.

- The set of invariant sentences is not closed under taking subformulae, even if we extend our definition to include invariant formulae with free variables. Thus we can not prove statements about invariant sentences by means of structural induction.
- No algebraic or game characterisations of the expressive power of invariant logics are known, except for those cases where it is implied by a collapse result in expressive power (such as  $<\text{-inv-FO} \equiv \text{FO}$  on certain trees, cf. Chapter 4).

We reproduce here the proof we gave in [EEH14, EEH17] that it is undecidable whether a given FO-sentence is order-invariant on the class of all star forests, i.e. graphs that are disjoint unions of graphs of the form  $K_{1,n}$  for some  $n \geq 0$ . A detailed proof that order-invariance for first-order logic on strings (with a successor relation) is undecidable has been given by Benedikt and Segoufin in [BS09, §3]. In contrast to this, order-invariance on  $\text{Fin}(\sigma)$  is decidable if the signature  $\sigma$  contains only unary relation symbols, a fact that has been mentioned in [Sch13]:

**Theorem 2.2.1.** *Let  $\sigma = \{P_1, \dots, P_k\}$  be a signature containing  $k$  unary relation symbols. Then given a sentence  $\varphi \in \text{FO}(\sigma^<)$  it is decidable whether  $\varphi$  is order-invariant on all finite  $\sigma$ -structures.*

*Proof.* We use the fact that a finite  $\sigma^<$ -structure  $A$  in which  $<$  is interpreted as a linear order may be viewed as a word  $w_A \in \Sigma^*$ , where  $\Sigma = 2^\sigma$  is the powerset of  $\sigma$ . Moreover, for a given  $\varphi \in \text{FO}(\sigma^<)$  the syntactic monoid of the language

$$L_\varphi = \{w_A \in \Sigma^* \mid A \models \varphi\}$$

defined by  $\varphi$  is finite and can be computed, cf. [Str94]. Now  $\varphi$  is order-invariant if, and only if, the syntactic monoid of  $L_\varphi$  is commutative, which is decidable.  $\square$

This argument can be extended to structures over signatures that may contain non-unary relation symbols, provided no two elements appear together in a relation (i.e. the Gaifman graph must be edgeless). The next theorem complements this result by showing that already in graphs of maximum degree 1 (i.e. partial matchings), order-invariance is no longer decidable.

**Theorem 2.2.2.** *There is a signature  $\sigma$  such that order-invariance for first-order sentences is undecidable on graphs of maximum degree 1.*



Our proof of this theorem uses a reduction from the undecidable halting problem for *counter machines* (cf. [Min67]) with two counters which store natural numbers. Such a machine executes a *programme*, i.e. a finite sequence  $P = I_1 \cdots I_\ell$  of instructions of the following types:

- $\text{INC}(i)$ : increase counter  $i \in \{1, 2\}$ , proceed with next instruction.
- $\text{DEC}(i, j_0, j_1)$ : if counter  $i$  is zero, proceed with instruction  $I_{j_0}$ . Otherwise decrease counter  $i$  and proceed with instruction  $I_{j_1}$ .
- $\text{HALT}$ : stop the execution.

For our purposes these machines can be assumed to start with empty input (i.e. both counters equal to zero), and since we are only interested in halting, we do not need to specify an acceptance condition. The configuration of the machine at any execution step is fully described by a triple  $(n_1, n_2, j)$ , where  $n_1, n_2 \geq 0$  are natural numbers stored in the counters and  $j \in [\ell]$  is the number of the next instruction to be executed. Without loss of generality, we assume that  $I_\ell$  is the unique  $\text{HALT}$  instruction in  $P$ . Hence we say that a program *halts* if, and only if, it reaches a configuration  $(n_1, n_2, \ell)$  for some  $n_1, n_2 \geq 0$  from the initial configuration  $(0, 0, 1)$ . Deciding whether a two-counter machine halts is undecidable (cf. [Min67]).

*Proof of Theorem 2.2.2.* The key observation used here (as well as in folklore proofs for undecidability of order-invariance, cf. [Lib04, Ex. 9.3]) is the fact that an undecidable sentence is necessarily order-invariant. We fix a signature  $\sigma = \{E, <, L_1, L_2, R_1, R_2, J, |\}$  with binary relation symbols  $E$  and  $<$  and unary relation symbols  $L_1, L_2, R_1, R_2, J$  and  $|$ .

For a given two-counter programme  $P$  we construct a sentence  $\varphi_P \in \text{FO}(\sigma)$  such that

- if  $P$  does not halt, then  $\varphi_P$  is not satisfiable (and therefore order-invariant), and
- if  $P$  does halt, then  $\varphi_P$  has a model which is linearly ordered by  $<$  and whose  $\{E\}$ -reduct is a partial matching, and  $\varphi_P$  is not order-invariant on partial matchings.

Thus a decision procedure for order-invariance on graphs of maximum degree 1 would yield a decision procedure for the halting problem of two-counter machines.

The idea behind the construction of  $\varphi_P$  is that its models encode halting computations of  $P$ . Let  $C_1, C_2, \dots, C_m$  be a sequence of configurations. We encode each configuration  $C = (n_1, n_2, j)$  as a string

$$\text{enc}(C) = (L_1 R_1)^{n_1} (L_2 R_2)^{n_2} J^j$$

and encode the sequence  $C_1, \dots, C_m$  as the concatenation

$$|\text{enc}(C_1)| \dots |\text{enc}(C_m)|$$

of these, with the letter  $|$  between any two parts and at both ends. As usual, we identify each non-empty word over the alphabet  $\{L_1, R_1, L_2, R_2, J, |\}$  with a  $\{<, L_1, R_1, L_2, R_2, J, |\}$ -structure. If a machine  $P$  halts after  $h$  steps, a word  $w_P \in \{L_1, R_1, L_2, R_2, J, |\}^*$  encodes the *run* of  $P$ , i.e. the finite sequence of configurations at time steps  $1, \dots, h$ , if

- (E1)  $w = |(L_1 R_1)^{n_1^{(1)}} (L_2 R_2)^{n_2^{(1)}} J^{j^{(1)}}| \dots |(L_1 R_1)^{n_1^{(t)}} (L_2 R_2)^{n_2^{(t)}} J^{j^{(t)}}|$  for suitable  $(n_1^{(t)}, n_2^{(t)}, j^{(t)})$ ,  $t = 1, \dots, h$ ,
- (E2) subsequent configurations encoded in  $w$  are in keeping with the programme  $P$ ,
- (E3)  $(n_1^{(1)}, n_2^{(1)}, j^{(1)}) = (0, 0, 1)$ , and
- (E4)  $w$  ends in  $J^\ell$  (i.e. the last configuration is halting).

Conditions (E1), (E3), and (E4) are easily checked in first-order logic. For (E2) we need to check that if

$$|(L_1 R_1)^{n_1} (L_2 R_2)^{n_2} J^j| (L_1 R_1)^{n'_1} (L_2 R_2)^{n'_2} J^{j'}|$$

is a subword of  $w$ , then

- (E2.1) if instruction  $I_j$  is of the form  $\text{INC}(i)$  then

$$n'_i = n_i + 1, \quad n'_{2-i} = n_{2-i}, \quad j' = j + 1.$$

- (E2.2) if  $I_j$  is of the form  $\text{DEC}(i, j_0, j_1)$  then  $n'_{2-i} = n_{2-i}$  and

$$(n_i = n'_i = 0 \text{ and } j' = j_0) \quad \text{or} \quad (n_i = n'_i + 1 > 0 \text{ and } j' = j_1),$$

and

- (E2.3) if  $I_j$  is a  $\text{HALT}$ -instruction, then  $n_1 = n'_1, n_2 = n'_2$ , and  $j' = j$ .

The conditions on  $j$  and  $j'$  can be explicitly checked in first-order logic because  $j$  and  $j'$  are bounded by  $\ell$ . To check the conditions relation  $n_i$  to  $n'_i$ , we expand  $w_P$  by interpreting the binary relation  $E$  with a matching between letters in the parts  $R_i^{n_i}$  and  $L_i^{n'_i}$ , possibly leaving one of them unmatched to account for the increase or decrease of a counter. Since the edges in this expansion form a partial matching, the Gaifman graph has maximum degree 1.

Using our description above, it is easy to write down a first-order sentence

$\varphi$  defining the class of all matching expansions of  $w_P$ . This class is non-empty iff  $P$  halts. Hence  $\varphi$  is satisfiable in the finite iff  $P$  halts, and  $\varphi$  is easily seen not to be order-invariant in case it has a finite model.  $\square$

## Chapter 3

# Structural Properties of Graphs

In this chapter we introduce several ways of measuring “tameness” of a relational structure. All of these will be introduced in graph theoretic terms, and when we say that a structure has one of these properties, we implicitly mean that its associated Gaifman graph (cf. Definition 1.1.5) has said property. The concepts defined here will be used in Parts II and III.

By a *graph* we mean a pair  $G = (V, E)$ , where  $V$  is a finite set whose elements are called *vertices* and

$$E \subseteq \binom{V}{2}$$

is a set of *edges*. We commonly omit set brackets and commas when writing edges, thus writing  $uv \in E$  rather than  $\{u, v\} \in E$ . Graphs in our sense are commonly referred to as finite undirected simple (i.e. without loops or parallel edges) graphs. We refer to Diestel’s book [Die12] for any notions not explicitly defined here.

A *path* in a graph  $G = (V, E)$  is a sequence

$$v_0, v_1, \dots, v_\ell \in V$$

of pairwise distinct vertices such that  $v_i v_{i+1} \in E$  for  $0 \leq i < \ell$ . The vertices  $v_0$  and  $v_\ell$  are called the *endpoints* of the path, and the path is said to *connect* its endpoints. The vertices  $v_1, \dots, v_{\ell-1}$  are called *internal vertices* of the path. Two paths are called *internally vertex-disjoint* if they share no internal vertex.

A graph  $G' = (V', E')$  is a *subgraph* of  $G = (V, E)$  if  $V' \subseteq V$  and  $E' \subseteq E$ .  
If

$$E' = E \cap \binom{V'}{2}$$

then  $G'$  is an *induced* subgraph. For a set  $X \subseteq V$  of vertices,

$$G[X] := \left(X, E \cap \binom{X}{2}\right)$$

is the subgraph of  $G$  induced by  $X$ .

For  $k \geq 1$  we denote by  $K_k$  the complete graph on  $k$  vertices, i.e.

$$K_k := ([k], \{ij \mid 1 \leq i < j \leq k\}),$$

and for a set  $M$  of vertices we denote by  $K[M]$  the complete graph with this vertex set:

$$K[M] := (M, \{uv \mid u, v \in M, u \neq v\}).$$

For  $m, n \in \mathbb{N}^+$  we denote by  $K_{m,n}$  the complete bipartite graph with  $m$  and  $n$  vertices, i.e.

$$K_{m,n} := (\{\ell_i \mid i \in [m]\} \cup \{r_i \mid i \in [n]\}, \{\ell_i r_j \mid i \in [m], j \in [n]\}).$$

A *separation* of a graph  $G = (V, E)$  is a pair  $(A, B)$  of nonempty subsets  $A, B \subseteq V$  such that  $V = A \cup B$  and there is no edge in  $G$  between any  $a \in A \setminus B$  and  $b \in B \setminus A$ . For  $c \in \mathbb{N}$ , a graph is called *c-connected* if there is no separation  $(A, B)$  with  $|A \cap B| < c$ . For  $c = 1$  we just say *connected*. The union of two graphs  $G = (V, E)$  and  $H = (W, F)$  is the graph

$$G \cup H := (V \cup W, E \cup F).$$

For this we require neither  $V$  and  $W$  nor  $E$  and  $F$  to be disjoint.

We let  $\sigma_G = \{E\}$  be a signature with just one binary relation symbol  $E$  and tacitly identify a graph  $G = (V, E)$  with the  $\sigma_G$ -structure  $A$  which has

$$V(A) = V \quad \text{and} \quad E^A = \{(u, v) \mid uv \in E\}.$$

The class of all structures of this form is denoted by **Graph**.

Let  $C$  be a set whose elements will be called *colours*. A *(vertex-)coloured* graph with colours in  $C$  is a graph  $G = (V, E)$  together with a function  $\lambda : V \rightarrow C$ . We call  $\lambda(v)$  the *colour* of  $v$ . Again we identify these graphs with  $\sigma_{G,C}$ -structures encoding them, where  $\sigma_{G,C} = \sigma_G \cup \{P_c \mid c \in C\}$ . The resulting class of structures is called  $\mathbf{Graph}_C$ .

### 3.1 Trees and Tree Decompositions

We begin by settling some notation and terminology:

**Definition 3.1.1** (Trees). A *tree* is a connected acyclic graph  $T = (V, E)$ . It is called *rooted* if there is a designated vertex  $r \in V$ , which is then called the *root* of  $T$ . In this case there is a natural partial order  $\preceq$  on  $V$  given by

$$u \preceq v \iff u = v \text{ or } u \text{ is on the unique path from } r \text{ to } v,$$

which is referred to as the *ancestor relation*. If  $u \preceq v$  we call  $u$  an *ancestor* of  $v$ . If, in addition, there is no  $w \in V \setminus \{u, v\}$  with  $u \preceq w \preceq v$  we call  $u$  the *parent* of  $v$  and  $v$  a *child* of  $u$ . Children of the same parent are called *siblings*. The *height* of a rooted tree is the maximum number of vertices (not edges) on a path from  $r$  to some vertex  $u$ ; in particular, a tree consisting of a single vertex has height 1. A *leaf* is a vertex  $u$  without children.

A class  $\mathcal{T}$  of trees is called *ranked* if there is a number  $k \in \mathbb{N}$  such that for every  $T = (V, E) \in \mathcal{T}$  and every  $u \in V$  the number of children is at most  $k$ . If no such bound exists the class  $\mathcal{T}$  is *unranked*.

A *siblinged* tree is a rooted tree  $T = (V, E)$  with a binary relation  $\preceq_s \subseteq V^2$  such that

- if  $u \preceq_s v$  then  $u$  and  $v$  are siblings, and
- for every  $v \in V$ ,  $\preceq_s$  is a successor relation on the set of children of  $v$ .

We sometimes treat the edges of rooted trees as directed away from the root, so the root node  $r$  is the unique node without incoming edges (whereas every other node has exactly one incoming edge).

**Tree Decompositions** A common strategy in algorithm design is called “divide and conquer”: In order to solve a problem, it is divided into instances of the same problem with smaller input size. The resulting instances are then solved and their results combined into a solution of the original problem. The process stops when the given instance is simple enough that the problem can be solved immediately (e.g. by brute force on a very small instance).

In light of the Feferman-Vaught-Theorem for first-order and monadic second-order logic (Theorem 1.1.16), we would like to repeatedly decompose the universe  $V$  of a structure into subsets  $A$  and  $B$  such that

- $|A \cap B|$  is small, and
- there is no edge  $uv$  with  $u \in A \setminus B$  and  $v \in B \setminus A$  in the Gaifman graph.

Repeating this process gives a tree structure, which is captured in the following definition:

**Definition 3.1.2.** A *tree decomposition* of a graph  $G = (V, E)$  is a tree  $\mathcal{T} = (T, F)$  and a family  $(\mathcal{V}_t)_{t \in T}$  of sets  $\mathcal{V}_t \subseteq V$  such that:

- (T1) Every  $v \in V$  is contained in at least one  $\mathcal{V}_t$ , and the subgraph induced by  $T_v := \{t \in T \mid v \in \mathcal{V}_t\}$  is connected in  $\mathcal{T}$ .
- (T2) For every edge  $uv \in E$  there is a  $t \in T$  such that  $\{u, v\} \subseteq \mathcal{V}_t$ .

In order to distinguish between vertices of the tree and vertices of the decomposed graph, the elements of  $T$  are commonly referred to as *nodes*. The sets  $\mathcal{V}_t$  are called *bags*.

The *torso*  $\bar{\mathcal{V}}_t$  of a bag  $\mathcal{V}_t$  is the graph induced by  $\mathcal{V}_t$  with edges added between any two vertices which are shared with the bag at a neighbour of  $t$ :

$$\bar{\mathcal{V}}_t := G[\mathcal{V}_t] \cup \bigcup_{st \in F} K[\mathcal{V}_s \cap \mathcal{V}_t]$$

The (*maximal*) *adhesion* of a tree decomposition is the maximum of  $|\mathcal{V}_s \cap \mathcal{V}_t|$  over all edges  $st \in F$ .

If  $\mathcal{T}$  is a path, the tree decomposition is called a *path decomposition*.

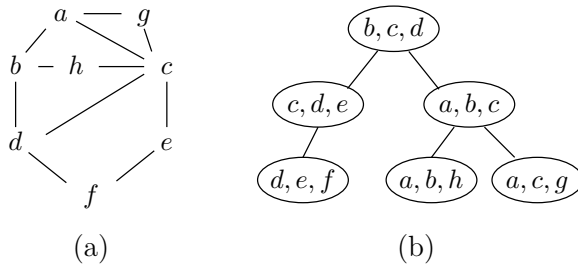


Figure 3.1: A sample graph (a) with a tree decomposition (b).

Figure 3.1 shows an example of a graph with a tree decomposition. Tree decompositions do indeed capture the process of repeatedly dividing the graph, in the following sense: Removing an edge  $st \in F$  from a tree  $\mathcal{T}$  separates the tree into exactly two components, one containing  $s$  and the

other one containing  $t$ . Denote these two components by  $\mathcal{T}_{s \vdash t}$  and  $\mathcal{T}_{t \vdash s}$ , respectively, and set

$$\mathcal{V}_{s \vdash t} := \bigcup_{r \in \mathcal{T}_{s \vdash t}} \mathcal{V}_r, \text{ and } \mathcal{V}_{t \vdash s} := \bigcup_{r \in \mathcal{T}_{t \vdash s}} \mathcal{V}_r.$$

Then every edge in  $\mathcal{T}$  yields a separator of the graph  $G$  as follows:

**Lemma 3.1.3.** *Let  $G = (V, E)$  be a graph,  $(\mathcal{T}, (\mathcal{V}_t)_{t \in T})$  a tree decomposition of  $G$  with  $\mathcal{T} = (T, F)$ ,  $st \in F$  and  $\mathcal{T}_{s \vdash t}$  etc. defined as above. Then  $\mathcal{V}_s \cap \mathcal{V}_t$  separates  $\mathcal{V}_{s \vdash t}$  from  $\mathcal{V}_{t \vdash s}$ .*

*Proof.* We first show that  $\mathcal{V}_{s \vdash t} \cap \mathcal{V}_{t \vdash s} \subseteq \mathcal{V}_s \cap \mathcal{V}_t$ . Indeed, if  $v \in \mathcal{V}_{s \vdash t} \cap \mathcal{V}_{t \vdash s}$ , then there are  $r \in \mathcal{T}_{s \vdash t}$  and  $r' \in \mathcal{T}_{t \vdash s}$  with  $v \in \mathcal{V}_r$  and  $v \in \mathcal{V}_{r'}$ . But both  $s$  and  $t$  lie on the unique path from  $r$  to  $r'$  in  $\mathcal{T}$ , so  $v \in \mathcal{V}_s \cap \mathcal{V}_t$  by (T1).

Now let  $v \in \mathcal{V}_{s \vdash t}$  and  $w \in \mathcal{V}_{t \vdash s}$  with  $vw \in E$ . We need to show that one of  $v$  and  $w$  is in  $\mathcal{V}_s \cap \mathcal{V}_t$ . By (T2) there is an  $r \in T$  with  $v, w \in \mathcal{V}_r$ . But if  $v \notin \mathcal{V}_t$ , then  $r \notin \mathcal{T}_{t \vdash s}$  by (T1), and if  $w \notin \mathcal{V}_t$ , then  $r \notin \mathcal{T}_{s \vdash t}$  by (T1).  $\square$

The *width* of such tree decomposition  $\mathcal{T}$  is defined as

$$\text{width}(\mathcal{T}) := \max\{|\mathcal{V}_t| \mid t \in T\} - 1.$$

The *treewidth*  $\text{tw}(G)$  of  $G$  is the minimal width of a tree decomposition of  $G$ . The graph in Figure 3.1(a) has treewidth at most 2, as witnessed by the decomposition given in Fig. 3.1(b).

A graph is called *d-degenerate* if every subgraph has a vertex of degree at most  $d$ .  $d$ -degenerate graphs are  $(d+1)$ -colourable, and such colourings can be computed efficiently using a greedy algorithm. Furthermore,  $d$ -degenerate graphs satisfy  $|E| \leq \frac{d}{2} |V|$ . The following lemma shows that every graph of treewidth at most  $k$  is  $k$ -degenerate.

**Lemma 3.1.4.** *Let  $G = (V, E)$  be a graph of treewidth at most  $k$ . Then  $G$  is  $k$ -degenerate.*

*Proof.* Since every subgraph of  $G$  again has treewidth at most  $k$  we only need to show that  $G$  has a vertex of degree at most  $k$ . Let  $(\mathcal{T}, (\mathcal{V}_t)_{t \in T})$  be a tree decomposition of  $G$  of width at most  $k$ . We may assume that for every tree edge  $st$  neither  $\mathcal{V}_s \subseteq \mathcal{V}_t$  nor  $\mathcal{V}_t \subseteq \mathcal{V}_s$ , since in these cases we may remove either  $s$  or  $t$  from  $\mathcal{T}$ . Now let  $t$  be a leaf of  $\mathcal{T}$ . Then there is a vertex  $v \in \mathcal{V}_t$  which is not contained in any other bag  $\mathcal{V}_s$ . Therefore all neighbours of  $v$



must be in  $\mathcal{V}_t$ , and since  $|\mathcal{V}_t| \leq k + 1$  it follows that  $v$  has degree at most  $k$ .  $\square$

Combined with the Feferman-Vaught-Theorem for monadic second-order logic (Thm. 1.1.16), Lemma 3.1.3 yields an efficient model checking algorithm for MSO on graphs of bounded treewidth, a seminal result going back to Courcelle. We review this in Chapter 6.

### 3.2 Treedepth

The *treedepth* of a graph may be defined similarly to treewidth by requiring the existence of a tree decomposition  $\mathcal{T}$  that simultaneously has low width and low height (as a tree). The following inductive definition is one of several equivalent formalisations of this concept (cf. [NOdM12] for a reference on treedepth):

$$\text{td}(G) := \begin{cases} 1 & \text{if } |V(G)| = 1 \\ 1 + \min_{r \in V(G)} \text{td}(G \setminus r) & \text{if } G \text{ is connected and } |V(G)| > 1 \\ \max_{i \in [n]} \text{td}(K_i) & \text{if } G \text{ has components } K_1, \dots, K_n. \end{cases}$$

This definition may be rephrased as saying that a depth-first search tree (cf. [AHU74, Sec. 5.1]) of height at most  $d$  exists, and can be found by cleverly selecting the order in which vertices are explored.

As usual, the treedepth  $\text{td}(A)$  of a relational structure  $A$  is defined as the treedepth of its Gaifman graph. We let

$$\mathbf{Fin}_\sigma^{\text{conn}} := \{A \in \mathbf{Fin}_\sigma \mid A \text{ is connected}\}$$

and for each  $d \in \mathbb{N}^+$ , we let

$$\begin{aligned} \text{TD}_d(\sigma) &:= \{A \in \mathbf{Fin}_\sigma \mid \text{td}(A) \leq d\}, \\ \text{TD}_d^{\text{conn}}(\sigma) &:= \{A \in \mathbf{Fin}_\sigma^{\text{conn}} \mid \text{td}(A) \leq d\}. \end{aligned}$$

As an immediate consequence of the above definition of treedepth, each  $A \in \text{TD}_d^{\text{conn}}(\sigma)$  with  $d > 1$  contains an element  $r$  with  $\text{td}(A \setminus r) < \text{td}(A)$ . This  $r$  is not uniquely determined. We call any vertex  $r$  such that  $\text{td}(A \setminus r) < \text{td}(A)$  a *treedepth root* and denote the set of all such vertices by  $\text{root}(A)$ ; these are exactly the vertices in which a depth-first search may start if it is

to produce a search tree of minimal height. By a result of [BDK12], the size of  $\text{root}(A)$  is bounded by a function of  $d$  (independent of the size of  $A$ ):

**Lemma 3.2.1** ([BDK12, Lem. 7]). *There is a function  $f : \mathbb{N}^+ \rightarrow \mathbb{N}^+$  such that  $|\text{root}(G)| \leq f(\text{td}(G))$  for each connected graph  $G$ .*

Note that the definition of  $\text{root}(G)$  in [BDK12] is slightly different from ours, but the two definitions are easily seen to be equivalent.

A graph of treedepth at most  $d$  can not contain a path of length  $2^d$  (cf. [NOdM12, 6.2]). Therefore  $\text{dist}_A(a, b) < 2^d$  for all elements  $a$  and  $b$  in the same connected component of a structure  $A$  of treedepth at most  $d$ , and the formula

$$\begin{aligned} \text{reach}_d(x, y) := & \exists x_1 \exists x_2 \dots \exists x_{2^d} ((x \dot{=} x_1 \vee E x x_1) \wedge \\ & (x_1 \dot{=} x_2 \vee E x_1 x_2) \wedge \dots \wedge (x_{2^d} \dot{=} y \vee E x_{2^d} y)) \end{aligned}$$

defines the reachability relation in these structures:

$$A \models \text{reach}_d[a, b] \quad \text{iff} \quad a \text{ and } b \text{ belong to the same component of } A.$$

We could have chosen a formula with  $d$  quantifier alternations and length  $O(d)$  equivalent to  $\text{reach}_d$  on graphs, rather than our existential formula with length  $\Theta(2^d)$ . However, we will use  $\text{reach}_d$  to relativise formulae  $\varphi(x)$  to the connected component of  $x$  by replacing subformulae

$$\begin{aligned} \exists z \psi \quad & \text{with} \quad \exists z (\text{reach}_d(x, z) \wedge \psi) \quad \text{and} \\ \forall z \psi \quad & \text{with} \quad \forall z (\neg \text{reach}_d(x, z) \vee \psi), \end{aligned}$$

so that the resulting formula  $\varphi|_{\text{reach}_d(x, z)}$  satisfies

$$A \models \varphi|_{\text{reach}_d(x, z)}[a] \quad \text{iff} \quad K \models \varphi[a],$$

where  $K$  is (the substructure of  $A$  induced on) the connected component of  $a$  in  $A$ . Then since  $\text{reach}_d$  is existential, we have

$$\text{qad}(\varphi|_{\text{reach}_d(x, z)}) = \text{qad}(\varphi).$$

Using these observations and the inductive definition of treedepth, it is easy to write down an  $\text{FO}(\sigma)$ -sentence that defines  $\text{TD}_d(\sigma)$  on the class of all finite  $\sigma$ -structures. While this naïve approach leads to a formula whose quantifier alternation depth grows linearly with  $d$ , it is also possible

to construct a *universal* sentence  $\text{td}_{\leq d}$  defining  $\text{TD}_d(\sigma)$  as a subclass of  $\text{Fin}_\sigma$ , cf. [NOdM12, Section 6.10] for details. Using this sentence, we construct a sentence that defines the set  $\text{root}(A)$  for each  $A \in \text{TD}_d^{\text{conn}}(\sigma)$  with  $d > 1$ . To this end, we let

$$\text{root}_d(x) := \bigvee_{c \leq d-1} (\text{td}_{>c} \wedge \text{td}_{\leq c|_{(x \neq z)}}(x)).$$

## Defining Bounded-Depth Tree Decompositions in FO

In this section we show how using Lemma 3.2.1 on the number of roots in a bounded treedepth-structure and the fact that connected components are FO-definable, for graphs of bounded treedepth, one can obtain an FO-interpretation of a bounded-depth tree decomposition. Furthermore, since the interpretation we give here is not parameterised we obtain a canonical tree decomposition, though not one of optimal depth or width. This construction is taken from our paper [EEH14, EEH17] and may be of independent interest. Note that it can be seen as an analogue, for bounded *treedepth* and FO, of Bojańczyk and Pilipczuk’s result [BP16] that tree decompositions of bounded width are definable in MSO.

The FO-interpretation is given by formulae  $\epsilon_d(x, y)$  and  $\alpha_d(x, y)$  for every  $d \geq 1$  such that if  $A$  is a  $\sigma$ -structure of treedepth at most  $d$  then

- $\epsilon_d$  defines an equivalence relation  $\sim_A := \{(u, v) \mid A \models \epsilon_d[u, v]\}$  on  $V(A)$ ,
- the equivalence classes of  $\sim_A$  have size bounded by a function of  $d$ ,
- the relation defined by  $\alpha_d$  is invariant under  $\sim_A$ , i.e. if  $u \sim_A u'$  and  $v \sim_A v'$ , then

$$A \models \alpha_d(u, v) \Leftrightarrow A \models \alpha_d(u', v'), \text{ and}$$

- $\alpha_d$  defines a rooted tree structure on  $V(A)/\sim_A$ , in which  $[u]_{\sim_A}$  is an ancestor of  $[v]_{\sim_A}$  or vice versa whenever  $u, v \in V(A)$  are adjacent in the Gaifman graph of  $A$ .

This can be turned into a bounded-depth tree decomposition in the usual sense by taking the tree structure on  $A/\sim_A$  as the tree and setting

$$\{v \mid [v]_{\sim_A} \text{ is an ancestor of } [u]_{\sim_A}\}$$

as the bag of the node  $[u]_{\sim_A}$ .

The key insight we use is Lemma 3.2.1 which says that for any fixed  $d$  there are at most  $f(d)$  many candidates which may be placed at the root of

a tree decomposition of  $G$  of minimum height. We have already seen above that there is an FO-formula  $\text{root}_d(x)$  such that  $A \models \text{root}_d[r]$  iff  $r$  is such a candidate. We recursively build a tree decomposition  $\mathcal{T}_G$  of  $G$  of height at most  $d$  by placing, in each step, *all* candidate roots into the root-bag of our tree decomposition and then recursing on the components of the remaining graph. Note that even if  $\text{td}(G) = d$ , not all components of  $G - R$ , where  $R$  is the set of at most  $f(d)$  root nodes, necessarily have treedepth  $d - 1$ , so we must be a bit careful which vertices we place into the root of the next level.

We fix a treedepth  $d$  and recursively define FO-formulae  $\varphi_i$  for  $i = 0, \dots, d$  with the intended meaning that, in a graph  $G = (V, E)$  of treedepth  $d$  with  $a \in V(G)$ ,  $G \models \varphi_i[a]$  iff  $a$  is on the  $i$ -th level of the tree decomposition, which we denote by  $L_i$ :

$$\begin{aligned}\varphi_0(x) &:= \perp \\ \varphi_i(x) &:= \bigvee_{j=1}^{d-i} \left( \text{td}_{=j+1} \mid \neg \varphi_{<i} \wedge \text{td}_{=j} \mid \neg(\varphi_{<i} \vee z \dot{=} x) \right)\end{aligned}$$

Here,  $x$  is the free variable of  $\varphi_i$  and  $z$  is the free variable of the formulae used in the restrictions. With the abbreviations

$$\varphi_{<i}(x) := \bigvee_{j<i} \varphi_j(x) \quad \text{and} \quad \varphi_{\leq i}(x) := \bigvee_{j \leq i} \varphi_j(x)$$

we define

$$\begin{aligned}\psi_0(x, y) &:= \top \\ \psi_{i+1}(x, y) &:= \text{reach}_{d-i+1} \mid \neg \varphi_{\leq i},\end{aligned}$$

i.e.  $\psi_i(u, v)$  holds iff  $u$  and  $v$  are in the same connected component of  $G - \bigcup_{j \leq i} L_j$ . We can now define an equivalence relation on  $G$  as follows:

$$\epsilon_d(x, y) := \bigvee_{1 \leq i \leq d} (\varphi_i(x) \wedge \varphi_i(y) \wedge \psi_i(x, y)),$$

i.e. two vertices are equivalent iff they appear on the same level of our tree decomposition and are in the same connected component of  $G$  after removing the levels above  $x$  and  $y$ . This is equivalent to saying that  $x$  and  $y$  appear in the same node of our tree decomposition.

Finally, we define tree edges (directed towards the root) by

$$\alpha_d(x, y) := \bigvee_{1 \leq i < d} (\varphi_i(x) \wedge \varphi_{i+1}(y) \wedge \exists u \exists v (Euv \wedge \epsilon(x, u) \wedge \psi_{i+1}(y, v))).$$

This construction is sketched in Figure 3.2.

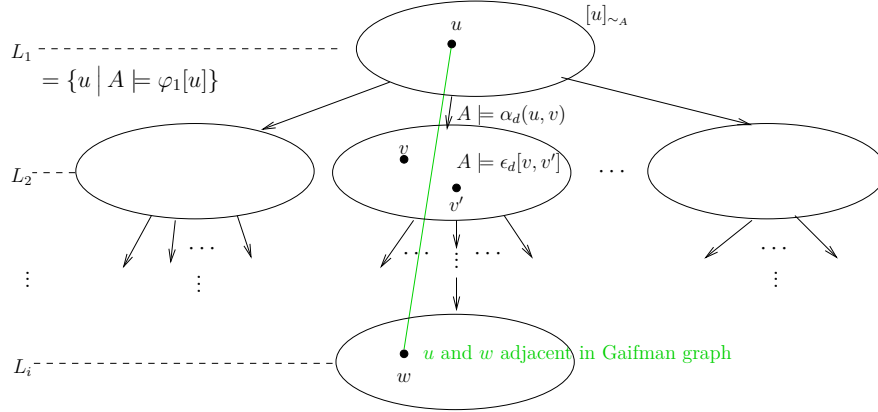


Figure 3.2: The canonical tree decomposition defined in FO.

### 3.3 Graphs of Bounded Genus

Some important classes of graphs are obtained by requiring them to be drawable on some surface, i.e. on a compact Hausdorff space locally homeomorphic to the plane, such that no two edges cross. We begin with some topological preliminaries.

**Definition 3.3.1.** A *surface* is a compact Hausdorff space  $\Sigma$  locally homeomorphic to the real plane  $\mathbb{R}^2$ , i.e. such that for every  $x \in \Sigma$  there is an open set  $x \in U \subseteq \Sigma$  and a bijective function  $\iota : U \rightarrow \mathbb{R}^2$  such that both  $\iota$  and  $\iota^{-1}$  are continuous.

The following classification theorem for surfaces is well known, a proof can be found in [Arm83, Chapter 7]:

**Theorem 3.3.2** (Classification of Compact Surfaces). *Every surface  $\Sigma$  is homeomorphic to*

1. *a sphere with a finite number of handles attached to it, or*
2. *a sphere with a finite number of cross caps attached to it.*

*Surfaces of the first kind are called orientable, those of the second kind are called non-orientable.*

**Definition 3.3.3** (curve,arc,loop). Let  $\Sigma$  be a surface or the plane  $\mathbb{R}^2$ . A *curve* on  $\Sigma$  is a continuous function  $\gamma : [0, 1] \rightarrow \Sigma$ . A curve  $\gamma$  on  $\mathbb{R}^2$  is called *polygonal* if there are finitely many values  $0 = x_0 < x_1 < \dots < x_k = 1$  such that

$$\gamma((1-t)x_{i-1} + tx_i) = (1-t)\gamma(x_{i-1}) + t\gamma(x_i)$$

for  $i \in [k]$  and  $t \in [0, 1]$ .

The points  $\gamma(0)$  and  $\gamma(1)$  are called *endpoints* of  $\gamma$ . Two points  $u, v \in X \subseteq \Sigma$  are called *connected* in  $X$  if there is a curve  $\gamma : [0, 1] \rightarrow X$  with endpoints  $u$  and  $v$ . This defines an equivalence relation on  $X$ , the equivalence classes of which are called *regions*. If  $X$  is an open subset of  $\Sigma$ , then also the regions of  $X$  are open. The *frontier* of a set  $X \subseteq \Sigma$  is the set  $\partial X$  of all points  $y \in \Sigma$  for which every neighbourhood of  $y$  intersects both  $X$  and its complement, i.e.

$$\partial X = \overline{X} \cap \overline{\Sigma \setminus X}.$$

A curve is called an *arc* if it is injective, with the possible exception of  $\gamma(0) = \gamma(1)$ , in which case it is called a *loop*. For an arc  $\gamma$ , the points  $\gamma(t)$  with  $t \in (0, 1)$  are called *interior points*. We will sometimes denote the image  $\{\gamma(t) \mid t \in [0, 1]\}$  by  $\gamma$  as well, and denote by  $\mathring{\gamma}$  the set  $\{\gamma(t) \mid t \in (0, 1)\}$  of interior points of  $\gamma$ . If  $\mathring{\gamma} \cap \mathring{\eta} = \emptyset$  for arcs  $\eta$  and  $\gamma$  we say that they are *internally disjoint*.

We will need the following theorem:

**Theorem 3.3.4** (Jordan Curve Theorem). *Let  $\gamma$  be a loop on  $\mathbb{R}^2$ . Then  $\mathbb{R}^2 \setminus \gamma$  has exactly two regions, exactly one of which is bounded, and  $\gamma$  is the frontier of both regions. In other words  $\mathbb{R}^2 = X \cup \gamma \cup Y$  with connected  $X$  and  $Y$ ,  $\partial X = \partial Y = \gamma$ , the union is disjoint and exactly one of  $X$  and  $Y$  is bounded.*

The Jordan Curve Theorem is actually true for arbitrary curves, but we will only need it for polygonal curves. Note that being connected by polygonal curves is a stronger equivalence relation than being connected by arbitrary (continuous) curves, but the theorem holds with both notions of

connectedness. A simple proof of this theorem for polygonal curves can be found in [MT01, Ch. 2].

We can now define what it means for a graph to be drawable on a surface:

**Definition 3.3.5.** A *drawing*  $\Pi$  of a graph  $G = (V, E)$  on a surface  $\Sigma$  associates a point  $\pi(v) \in \Sigma$  to every  $v \in V$  and an arc  $\gamma := \pi(e)$  to every edge  $e = uv$  such that  $\gamma(0) = \pi(u)$  and  $\gamma(1) = \pi(v)$  or the other way around and such that no two such paths share an interior point (i.e., a point  $\gamma(x)$  for some  $0 < x < 1$ ), and no interior point equals  $\pi(v)$  for some  $v \in V$ . A graph  $G$  is called *embeddable* into some surface  $\Sigma$  iff such a drawing exists. A graph which is embeddable into the sphere is called *planar*.

A *face* of a drawing  $\Pi$  is a region of  $\Sigma - \Pi$ , where we identify  $\Pi$  with the subset

$$\{\pi(v) \mid v \in V\} \cup \bigcup_{e \in E} \pi(e)$$

of  $\Sigma$ . The set of all faces of  $\Pi$  is denoted by  $\mathcal{F}(\Pi)$ .

A drawing is called *cellular* if every face is homeomorphic to an open disc. Cellular drawings can be described combinatorially by so-called *2-cell embedding schemes*, cf. [MT01, Ch. 3]. When we speak of a 2-cell embedding  $(G, \Pi)$  of a graph  $G$  in an algorithmic context, we mean a suitable representation of such an embedding scheme.

## 3.4 Minors and Topological Subgraphs

For two graphs  $G$  and  $H$  there are several natural notions of what it means for  $H$  to be contained in  $G$ , such as  $H$  being (isomorphic to) a subgraph or an induced subgraph of  $G$ . We introduce the concepts of minors and topological subgraphs.

### Minors

**Definition 3.4.1** (contractions, minors). Let  $G = (V, E)$  be a graph, and  $e = uv \in E$  an edge of  $G$ . The graph  $G/e = (V', E')$  is defined by

$$\begin{aligned} V' &:= (V \setminus \{u, v\}) \cup \{v_e\} \quad \text{for a new vertex } v_e, \\ E' &:= \left( E \cap \binom{V'}{2} \right) \cup \{v_e w \mid uw \in E \text{ or } vw \in E\}. \end{aligned}$$

The operation taking  $G$  to  $G/e$  is called *contraction* of the edge  $e$ .

A graph  $G'$  is called a *minor* of  $G$ , written  $G' \preceq G$ , iff  $G'$  is isomorphic to a graph obtained from  $G$  by taking a subgraph and contracting edges.

An equivalent characterisation of the minor relation is as follows: Let  $H = (W, F)$  and  $G = (V, E)$  be graphs. Then  $H \preceq G$  if for every  $w \in W$  there is a nonempty connected subgraph  $U_w$  of  $G$  such that

- $U_{w_1} \cap U_{w_2} = \emptyset$  for  $w_1 \neq w_2$ , and
- if  $w_1 w_2 \in F$  then there are  $v_1 \in U_{w_1}$  and  $v_2 \in U_{w_2}$  such that  $v_1 v_2 \in E$ .

The sets  $U_w$  together with some choice of edges  $v_i v_j$  as above are said to be an *image* of  $H$  in  $G$ .

Planar graphs and graphs of bounded treewidth are two examples of classes of graphs which are closed under taking minors, i.e. if  $H$  is a minor of a planar graph, then it is again planar, and if  $H$  is a minor of a graph of treewidth  $k$  then the treewidth of  $H$  is at most  $k$ . There is a rich structure theory for classes of graphs which are closed under taking minors, which can be used for algorithmic purposes. We need a few preparations.

Let  $k \geq 0$  and let  $\Sigma$  be a surface with disjoint closed discs  $D_1, \dots, D_k \subseteq \Sigma$ . Up to homeomorphism, the space  $\Sigma \setminus \bigcup_i \text{int}(D_i)$  only depends on  $k$  and  $\Sigma$  (and not, say, on the positions of the discs  $D_i$ ; cf. [Arm83, Ch. 7]), and we denote it by  $\Sigma - k$ . For  $i = 1, \dots, k$ , let  $f_i : [0, 1] \rightarrow D_i$  be a loop which follows the boundary curve  $C_i$  of  $D_i$ . Following Diestel [Die12], we call the  $C_i$  the *cuffs* of  $\Sigma - k$ , and  $f_i(0) \in \Sigma$  the *root* of  $C_i$ . On each cuff  $C_i$  the loop  $f_i$  induces a linear order on the points of  $C_i$ .

For a graph  $G = (V, E)$ , a  $s \geq 0$  and a surface  $\Sigma$ , we say that  $G$  is *s-nearly embeddable into*  $\Sigma$  if there is a set  $X \subseteq V$  with  $|X| \leq s$  such that  $G - X$  can be written as  $H_0 \cup H_1 \cup \dots \cup H_s$  so that

- (a) there is a drawing  $\Pi$  of  $H_0$  on  $\Sigma - s$  that meets cuffs only in vertices and which does not meet the root of a cuff,
- (b) the (possibly empty) graphs  $H_1, \dots, H_s$  are pairwise disjoint and common vertices of  $H_0$  and  $H_i$  are exactly those vertices of  $H_0$  which are mapped to  $C_i$  by  $\Pi$ ,
- (c) for  $i = 1, \dots, s$ , the graph  $H_i$  has a path decomposition  $(\mathcal{P}, \mathcal{V})$  of width  $< s$  such that the vertices of  $\mathcal{P}$  are the vertices of  $H_0 \cap H_i$ , ordered in the linear order of the cuff  $C_i$ , and  $v \in \mathcal{V}_v$  for these vertices. The graphs  $H_i$  are called *vortices* attached to  $H_0$ .

The vertices in  $X$  are called *apices* of the  $s$ -near embedding.



**Theorem 3.4.2** (Graph Structure Theorem). *Let  $G$  be a graph that does not contain  $K_r$ . Then there is a tree decomposition of  $G$  such that the torsos of all of its bags are  $s$ -nearly embeddable into some surface into which  $K_r$  is not embeddable, for some  $s$  depending only on  $r$ . Moreover, such a decomposition can be computed in fixed-parameter tractable time.*

For a proof that such a decomposition exists cf. [Die12, Thm. 12.4.11]. and for a fixed-parameter tractable algorithm computing it cf. [GKR13, DHK05]). This allows many proofs and algorithmic techniques for graphs planar graphs to be generalised to arbitrary classes of graphs with excluded minors, by first computing a tree decomposition into nearly embeddable graphs which can then be turned into planar graphs by removing cycles from them.

In Chapter 7 we will add edges to the graphs inside the bags of a tree decomposition into nearly embeddable subgraphs. The following lemma will allow us to conclude that unless we create large cliques inside one of the bags, there will be no substantially larger cliques in the whole graph:

**Lemma 3.4.3.** *Let  $G = (V, E)$  be a graph,  $k \in \mathbb{N}$ , and let  $(A, B)$  be a separation of  $G$  with  $|A \cap B| \leq k$  and such that  $G[A \cap B]$  is a clique. If  $K_{k+1} \preceq G$  then  $K_{k+1} \preceq A$  or  $K_{k+1} \preceq B$ .*

*Proof.* Suppose  $K_{k+1} \preceq G$  and let  $X_1, \dots, X_{k+1}$  be disjoint nonempty connected subgraphs of  $G$  witnessing this (i.e., such that there is an edge in  $G$  between some  $u \in X_i$  and  $v \in X_j$  for every  $1 \leq i < j \leq k+1$ .) Since  $|A \cap B| \leq k$ , some  $X_i$  does not meet this set, and so must lie entirely on one side of the separation. Without loss of generality we assume  $X_1 \subseteq A \setminus B$ .

Since there are no outgoing edges from  $X_1$  to  $B \setminus A$  we have  $X_j \cap A \neq \emptyset$  for  $j = 2, \dots, k+1$ . Since  $A \cap B$  is a clique we may replace  $X_j$  by  $X_j \cap A$ , thus  $K_{k+1} \preceq A$ .  $\square$

## Topological Subgraphs

**Definition 3.4.4** (topological subgraph). Let  $G = (V, E)$  and  $H = (W, F)$  be two graphs. We say that  $H$  is a *topological subgraph* or *topological minor* of  $G$  (written  $H \preceq_{\text{top}} G$ ) if there is an injective function  $\iota : W \rightarrow V$  and for every edge  $uv \in F$  a path in  $G$  connecting  $\iota(u)$  and  $\iota(v)$  such that the paths corresponding to different edges of  $H$  are internally vertex-disjoint. The subgraph of  $G$  consisting of these paths is called a *topological minor image* of  $H$  in  $G$ .

Obviously, if  $H$  is a topological subgraph of  $G$ , then it is also a minor, i.e.

$$H \preceq_{\text{top}} G \quad \Rightarrow \quad H \preceq G,$$

because we can contract the edges in a topological minor image of  $H$  in  $G$ . Therefore if  $\mathcal{G}$  is a class of graphs excluding some graph  $H$  as a minor, then it also excludes the same graph as a topological subgraph. The converse is not true, however: The class of all cubic graphs (i.e. graphs in which every vertex has degree three) excludes  $K_5$  as a topological subgraph, but for every graph  $H$  there is a cubic graph  $G$  such that  $H \preceq G$ .

Grohe and Marx [GM15] showed the following structure theorem for graphs with excluded topological subgraphs:

**Theorem 3.4.5** (Theorem 4.1 in [GM15]). *For every  $k \in \mathbb{N}$  there exists a constant  $c = c(k) \in \mathbb{N}$  such that the following holds: If  $H$  is a graph on  $k$  vertices and  $G$  a graph which does not contain  $H$  as a topological subgraph, then there is a tree decomposition  $(\mathcal{T}, \mathcal{V})$  of  $G$  of adhesion at most  $c$  such that for all  $t \in \mathcal{T}$*

- $\bar{\mathcal{V}}_t$  has at most  $c$  vertices of degree larger than  $c$ , or
- $\bar{\mathcal{V}}_t$  excludes  $K_c$  as a minor.

*Furthermore, there is an algorithm that, given graphs  $G$  of size  $n$  and  $H$  of size  $k$  computes such a decomposition in time  $f(k) \cdot n^{O(1)}$  for some computable function  $f : \mathbb{N} \rightarrow \mathbb{N}$ .*



## Part II

# Expressivity and Succinctness



## Chapter 4

# Variants of First-Order Logic

In this chapter we investigate the expressive power and relative succinctness of first-order logic with an invariantly used linear order ( $<$ -inv-FO) or successor relation (succ-inv-FO).

We start by reviewing some known examples of queries definable in  $<$ -inv-FO but not in FO, and a similar query separating succ-inv-FO from FO. All of these are graph-theoretically quite complex: They contain subgraphs or minors isomorphic to large cliques as well as large ladders, i.e. bipartite graphs of the form

$$V = \{u_i, v_i \mid i \in [n]\} \quad \text{and} \quad E = \{u_i v_j \mid 1 \leq i \leq j \leq n\}.$$

In Section 4.2 we first review known collapse results for  $<$ -inv-FO on certain kinds of trees and slightly extend these to siblinged unranked trees of bounded depth (this is unpublished work together with Anuj Dawar). On structures of bounded treedepth, we obtain very strong quantitative collapse results, summarised in Figure 4.1. These are obtained also for MSO and  $<$ -inv-MSO and presented in Chapter 5.

The gap in complexity between structures where  $<$ -inv-FO is known to be stronger than FO and those where collapse results are known is enormous, and has been the object of intensive study in recent years. We review some known results about locality properties of  $<$ -inv-FO in Section 4.3, as these might turn out to be useful for proving collapse results on further classes of structures.

$\varphi \in$	$<\text{-inv-FO}$	MSO	$<\text{-inv-MSO}$
$\psi \in$	FO	FO	FO <sub>mod</sub>
$\ \psi\ $	$d\text{-EXP}(q)$	$d\text{-EXP}(q)$	non-elementary
$\text{qad } \psi$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$

Figure 4.1: Summary of our results on  $<\text{-inv-FO}$  and  $<\text{-inv-MSO}$  on structures of bounded treedepth: A formula  $\varphi$  of quantifier rank  $q$  is translated into a formula  $\psi$  that is equivalent to  $\varphi$  on structures of treedepth at most  $d$ . The results for MSO and  $<\text{-inv-MSO}$  are presented in Chapter 5.

## 4.1 Separation Results

There are several examples of queries definable in  $<\text{-inv-FO}$  but not in FO. The first is an unpublished result usually attributed to Gurevich:

**Theorem 4.1.1** (Gurevich, unpublished). *The class of Boolean algebras with an even number of atoms is axiomatisable in  $<\text{-inv-FO}$  but not in FO.*

*Proof.* Let  $\sigma = \{\subseteq\}$  be the signature consisting only of the binary relation symbol  $\subseteq$ . It is well known that Boolean algebras can be axiomatised in FO by stating that

- $\subseteq$  is the partial order of a distributive lattice,
- this lattice has a maximal element 1 and a minimal element 0,
- every element  $a$  has a complement  $a'$  such that

$$\begin{aligned} (a \vee b)' &= a' \wedge b', & (a \wedge b)' &= a' \vee b', \\ a \vee a' &= 1, \text{ and} & a \wedge a' &= 0. \end{aligned}$$

Atoms are then elements  $x \neq 0$  such that if  $0 \subseteq y \subseteq x$  then  $y = 0$  or  $y = x$ . If  $X$  is the set of atoms of a finite Boolean algebra, then this algebra is isomorphic to  $(2^X, \subseteq)$ . For details, cf. [Cam94, Ch. 12].

A linear  $\leq$  order on a set  $S$  induces a linear order  $\leq \cap (T \times T)$  on each non-empty subset  $T \subseteq S$ . In particular, we can say that a linearly ordered Boolean algebra has an even number of atoms by saying there is a set which contains the first atom, then every other one, and the last atom is not contained in it. On the other hand, using Ehrenfeucht-Fraïssé-games it is easy to show that this class is not elementary, i.e. not axiomatisable in FO alone.  $\square$

Another example of a query definable in  $<\text{-inv-FO}$  but not in FO is the

class of structures given by Otto in [Ott00] to separate  $\epsilon$ -FO from FO on finite structures. We already discussed in Section 2.1 that this also separates  $<$ -inv-FO from FO.

A third query definable in  $<$ -inv-FO but not in FO is given by Potthoff in [Pot94]: He shows that the class of all full unordered binary trees of even height is definable in  $<$ -inv-FO but not in FO. In this query, trees are encoded as structures with not only the edge-relation but also its transitive closure  $\preceq$  (the ancestor relation), so this result does not contradict the results about trees which we present in Section 4.2.

**succ-inv-FO on finite structures** The known queries separating  $<$ -inv-FO from FO rely on the fact that a linear order on a structure induces a linear order on every subsets of the structure. While a successor relation on a structure uniquely determines a successor relation on each subset, this successor relation is not, in general, definable in FO. The reason is that the distance, in terms of the number of steps in the successor relation, between two consecutive elements of a subset may be arbitrarily large, and it is well known that FO can not express the transitive closure of a relation.

In [Ros07], Rossman gives an example of a query definable in **succ-inv-FO** but not in FO. Starting with Gurevich's query of Boolean algebras with an even number of atoms, Rossman adds further information to the structures which allow a successor relation on the set of atoms of the algebra to be defined in FO using a successor relation on the whole structure. In particular, the structures in this query also contain large cliques and large ladders.

## 4.2 Trees

In contrast to the results presented in the previous section we now present results which show that on certain restricted classes of structures order-invariant logics gain no expressive power over their plain counterparts. We review known results by Benedikt and Segoufin for various classes of trees and show how they can be extended.

### Unranked or Unsiblinged Trees

Recall that by Definition 3.1.1, a class of rooted trees is called *ranked* if there is a bound on the number of children of a vertex, and it is called *siblinged* if there is a binary relation  $\preceq_s$  which is a successor relation when restricted to the set of children of some vertex.



In [BS05], Benedikt and Segoufin give an algebraic characterisation of those classes of ranked *or* siblinged trees that are definable in FO. In particular they show that for every  $k \geq 1$  there exists a quantifier rank  $q = q(k)$  such that given any two trees  $T_1, T_2$  that are  $\text{FO}_q$ -equivalent, one can transform  $T_1$  into  $T_2$  using a sequence of operations called

- $k$ -guarded horizontal swaps,
- $k$ -guarded vertical swaps, and
- replacing a tree of the form  $D \cdot C^{k+1} \cdot t$  by  $D \cdot C^k \cdot t$ , where  $C$  and  $D$  are *contexts*, i.e. trees with a designated leaf which may be identified with the root of another context or tree.

We do not present the details here, but the classification of the expressive power of FO and the algebraic tools used to obtain it are similar to those for FO on words with a successor relation, cf. [Str94, VI.3]. Note that words with a successor relation can be seen as degenerate labelled trees, with 0 or 1 child per vertex.

Using this result, Benedikt and Segoufin show in [BS09] that  $<\text{-inv-FO} \equiv \text{FO}$  on unranked unsiblinged trees and on ranked trees (siblinged or not). Translated into the terminology we use in the next subsection, they show that if two trees  $T_1, T_2$  can be transformed into one another by one of the three operations stated above, then they are *equivalently orderable*, i.e.  $T_1 \leftrightarrow_q T_2$  for some quantifier rank  $q$ .

## Siblinged Trees of Bounded Depth

While Benedikt and Segoufin proved that  $<\text{-inv-FO} \equiv \text{FO}$  on unranked unsiblinged trees, the question of whether  $<\text{-inv-FO} \equiv \text{FO}$  on *siblinged* unranked trees remains open. We prove that  $<\text{-inv-FO} \equiv \text{FO}$  on siblinged unranked trees of bounded depth, i.e. for every  $k \geq 1$  and every  $\varphi \in <\text{-inv-FO}$  there is a  $\varphi_k \in \text{FO}$  such that

$$T \models \varphi \quad \Leftrightarrow \quad T \models \varphi_k$$

for every siblinged tree  $T$  of depth at most  $k$ . Note that because of the sibling relation these trees do *not* have treedepth bounded by any constant.

The results in this section were obtained in collaboration with Anuj Dawar in 2014 and have not yet been published. By induction on the height of the trees we prove a slightly stronger statement.

**Definition 4.2.1.** Let  $q \geq 0$ ,  $\sigma$  a signature and  $<$  a binary relation symbol not contained in  $\sigma$ . We say that two (finite)  $\sigma$ -structures  $A$  and  $B$  are  $q$ -

equivalently orderable, written  $A \leftrightarrow_q B$ , if there are linear orders  $<^A$  and  $<^B$  such that

$$(A, <^A) \equiv_q (B, <^B).$$

Note that  $\leftrightarrow_q$  is obviously reflexive and symmetric, but not necessarily transitive in general. Denote by  $\leftrightarrow_q^*$  the transitive closure of this relation. Since  $A \leftrightarrow_q B$  implies that  $A$  and  $B$  are  $<\text{-inv-FO}_q$ -equivalent, which is a transitive relation, we see that also

$$A \leftrightarrow_q^* B \Rightarrow A \equiv_{<\text{-inv-FO}} B.$$

We will show the following:

**Theorem 4.2.2.** *For every  $h \geq 1$  and  $q \geq 0$  there is a  $q'_h$  such that*

$$A \equiv_{q'_h} B \quad \Rightarrow \quad A \leftrightarrow_q^* B,$$

*for all siblinged unranked trees  $A, B$  of height at most  $h$ .*

This implies that for every  $h \geq 1$  and every  $\varphi \in <\text{-inv-FO}$  of quantifier rank  $q$  there is a  $\varphi_h \in \text{FO}$  of quantifier rank at most  $q_h$  such that  $\varphi \equiv \varphi_h$  on siblinged unranked trees of height at most  $h$ .

Using a variant of the Feferman-Vaught Theorem for ordered sums, one easily obtains the following lemma:

**Lemma 4.2.3.** *Let  $A_1 \leftrightarrow_q^* A_2$ , and  $B_1 \leftrightarrow_q^* B_2$ , and assume that  $V(A_i)$  and  $V(B_i)$  are disjoint for  $i = 1, 2$ . Then also*

$$A_1 \sqcup B_1 \leftrightarrow_q^* A_2 \sqcup B_2.$$

The proof of Theorem 4.2.2 proceeds by induction on  $h$ . We will use the following lemma, which follows from Benedikt and Segoufin's proof [BS09]:

**Lemma 4.2.4.** *For every quantifier rank  $q$  and every alphabet  $\Sigma$  there is a  $q'$  such that if two strings  $w_1, w_2 \in \Sigma^*$  are  $\equiv_{q'}$ -equivalent, then  $w_1 \leftrightarrow_q^* w_2$ . Here, we treat the strings as structures with one successor relation and unary predicates for the elements of the alphabet.*

*Proof of Thm. 4.2.2.* We assume by induction on  $h$  that every two  $h$ -SUTs which are  $\equiv_{q'_h}$ -equivalent are also  $\leftrightarrow_q^*$ -equivalent. Pick one tree  $T_C$  from each  $\equiv_{q'_h}$ -equivalence class  $\mathcal{C}$  of such trees. Then in particular  $T \leftrightarrow_q^* T_C$  for every  $T \in \mathcal{C}$ .

Choose  $q_{h+1}'$  large enough so that for any two  $(h+1)$ -SUTs  $T_1$  and  $T_2$ , if  $T_1 \equiv^{q_{h+1}'} T_2$  then the strings formed by the  $\equiv^{q_h}$ -types of the level 1 subtrees of  $T_1$  and  $T_2$  are  $q'$ -equivalent, where  $q'$  is chosen as in Lemma 4.2.4 for the alphabet  $\Sigma$  which consists of all  $\equiv^{q_h}$ -types of  $h$ -SUTs. This  $q_{h+1}'$  depends only on  $q$  and  $h$ , because  $q'$  and (by induction)  $q_h'$  do.

Let  $T_1'$  and  $T_2'$  be the trees obtained from  $T_1$  and  $T_2$  by replacing each level 1 subtree by the canonical representative  $\tilde{T}$  of its  $\equiv^{q_h}$ -equivalence class. Then  $T_1 \leftrightarrow_q^* T_1'$  and  $T_2 \leftrightarrow_q^* T_2'$  by repeated application of Lemma 4.2.3 and the induction hypothesis. Note that we may turn the structures into *disjoint* unions of a subtree rooted at a child of the root and the remaining tree by colouring the (at most three) nodes with connections to the root of the subtree in new colours.

Since all level 1 subtrees in  $T_1'$  and  $T_2'$  are canonical representatives of their respective  $\equiv^{q_h}$ -classes, the structures  $T_1'$  and  $T_2'$  may be interpreted by an FO-interpretation within strings  $w_1$  and  $w_2$  whose letters are the  $\equiv^{q_h}$ -equivalence classes of the level 1 subtrees of  $T_1$  and  $T_2$ . We choose  $\tilde{q}$  large enough that for every  $\text{FO}_q$ -sentence  $\varphi$  there is an  $\text{FO}_{\tilde{q}}$ -sentence  $\psi$  such that

$$T_i' \models \varphi \quad \Leftrightarrow \quad w_i \models \psi$$

for  $i = 1, 2$ . In fact, we may extend this to expansions of  $w_i$  with linear orders: For every linear order  $<$  on  $w_i$ , there is a linear order  $<'$  on  $T_i'$  such that

$$(T_i', <') \models \varphi \quad \Leftrightarrow \quad (w_i, <) \models \psi.$$

Using Lemma 4.2.4 we get

$$w_1 \leftrightarrow_{\tilde{q}}^* w_2,$$

and this allows us to conclude that

$$T_1' \leftrightarrow_q^* T_2'.$$

But now

$$T_1 \leftrightarrow_q^* T_1' \leftrightarrow_q^* T_2' \leftrightarrow_q^* T_2$$

as was to be proved. □

### 4.3 Locality

We review known locality properties of first-order logic. Originally going back to Gaifman [Gai82] they can also be found (with proofs different from Gaifman's original one) in textbooks such as [EF99, Lib04].

Recall that the Gaifman graph  $G_A$  of a relational structure  $A$  has vertex set  $V(A)$  and an edge between two elements if they appear together in a tuple of some relation.

**Definition 4.3.1** ( $r$ -neighbourhood). For an element  $a \in V(A)$  and a radius  $r \in \mathbb{N}$  we define the  $r$ -neighbourhood  $N_A^r(a)$  recursively as

$$\begin{aligned} N_A^0(a) &:= \{a\} \\ N_A^{r+1}(a) &:= N_A^r(a) \cup \{b \in V(A) \mid \text{there is a } c \in N_A^r(a) \text{ with } bc \in E\}, \end{aligned}$$

where  $E$  is the edge set of the Gaifman graph  $G_A$  of  $A$ . For a tuple  $\bar{a} = (a_1, \dots, a_k)$  of elements of  $V(A)$  we set

$$N_A^r(\bar{a}) := \bigcup_i N_A^r(a_i).$$

We drop the subscript  $A$  if the structure  $A$  is clear from the context. If  $A$  is a relational structure then  $N_A^r(a)$  is the universe of an induced substructure of  $A$ , which we also call  $N_A^r(a)$ , and similarly for tuples  $\bar{a}$ .

When we say that

$$N_A^r(\bar{a}), \bar{a} \simeq N_B^r(\bar{b}), \bar{b}$$

for two tuples  $\bar{a} \in V(A)^k$  and  $\bar{b} \in V(B)^k$  in  $\sigma$ -structures  $A$  and  $B$ , we mean that there is an isomorphism between the structures induced on the two neighbourhoods which maps  $a_i$  to  $b_i$  for  $i = 1, \dots, k$ .

It is easy to construct FO-formulae  $d(\bar{x}, y) < r$  and  $d(\bar{x}, y) \geq r$  such that, in a structure  $A$  with elements  $\bar{a}$  and  $b$ ,

$$A, \bar{a}, b \models d(\bar{x}, y) < r \quad \text{if, and only if,} \quad b \in N_A^{r-1}(\bar{a})$$

and

$$A, \bar{a}, b \models d(\bar{x}, y) \geq r \quad \text{if, and only if,} \quad b \notin N_A^{r-1}(\bar{a}).$$

**Definition 4.3.2** (local formulae). Let  $\sigma$  be a relational signature,  $\varphi \in \text{FO}(\sigma)$  a formula with free variables among  $\{x_1, \dots, x_k\}$ , and  $r \in \mathbb{N}$ . We call  $\varphi$  (*semantically*) *r-local* if

$$A \models \varphi[a_1, \dots, a_k] \quad \text{if, and only if,} \quad N_A^r(\bar{a}) \models \varphi[a_1, \dots, a_k]$$

for every  $\sigma$ -structure  $A$  and elements  $a_1, \dots, a_k \in V(A)$ .

We call  $\varphi$  *syntactically r-local* if all quantifications in it are of the form

$$\exists y (d(\bar{x}, y) < r \wedge \psi) \quad \text{or} \quad \forall y (d(\bar{x}, y) \geq r \vee \psi),$$

where  $y$  is a variable distinct from  $x_1, \dots, x_k$ .

A *basic r-local sentence* is a formula of the form

$$\exists x_1 \dots \exists x_k \left( \bigwedge_i \psi^{(r)}(x_i) \wedge \bigwedge_{i < j} d(x_i, x_j) > 2r \right),$$

where  $\psi^{(r)}(x_i)$  is a syntactically *r-local* formula around  $x_i$  (obtained from a common formula  $\psi^{(r)}(x)$  by substituting  $x_i$  for  $x$ ). The number  $k$  is the *width* of the basic local sentence.

Checking whether a formula is semantically local is easily seen to be undecidable, but syntactically local formulae are also semantically local. Gaifman [Gai82] proved the following result:

**Theorem 4.3.3.** *For every first-order logical formula  $\varphi(\bar{x})$  there are  $r, s, t \in \mathbb{N}$  such that  $\varphi$  is equivalent to a Boolean combination of basic *r-local* sentences of width *s* and formulae  $\psi^{(t)}(\bar{x})$  which are *t-local* around  $\bar{x}$ .*

*In particular, if  $\varphi$  is a sentence, then it is equivalent to a Boolean combination of basic *r-local* sentences.*

*If  $\varphi$  has quantifier rank  $q$  and  $m$  free variables (i.e.  $\bar{x} = (x_1, \dots, x_m)$ ), then we can get*

$$r \leq 7^{q-1}, \quad s \leq m + q, \quad \text{and} \quad t \leq \frac{1}{2}(7^q - 1).$$

*Furthermore, the translation from  $\varphi$  to this Boolean combination is computable.*

As a corollary, we obtain:

**Corollary 4.3.4.** *For every  $\varphi(x_1, \dots, x_k) \in \text{FO}$  there is an  $r \in \mathbb{N}$  such that if  $\bar{a}, \bar{b} \in V(A)^k$  are tuples in some structure  $A$  such that*

$$N_A^r(\bar{a}) \simeq N_A^r(\bar{b}),$$

*then*

$$A \models \varphi[\bar{a}] \quad \text{if, and only if,} \quad A \models \varphi[\bar{b}].$$

This corollary also holds for *order-invariant* first-order logic, as shown by Grohe and Schwentick [GS00]. Technically, we defined  $<\text{-inv-FO}$  as a set of *sentences*, and to prove locality in a structure  $A$  we only need order-invariance on this structure, so we spell out the invariance condition explicitly:

**Theorem 4.3.5.** *Let  $\varphi(x_1, \dots, x_k) \in \text{FO}_q(\sigma^<)$  be order-invariant on some  $\sigma$ -structure  $A$ . Then there is an  $r \in \mathbb{N}$  depending only on  $k, \sigma$  and the quantifier rank of  $\varphi$  such that if*

$$N_A^r(\bar{a}) \simeq N_A^r(\bar{b})$$

*for tuples  $\bar{a}, \bar{b} \in V(A)^k$ , then*

$$A' \models \varphi[\bar{a}] \quad \text{if, and only if,} \quad A' \models \varphi[\bar{b}],$$

*where  $A'$  is the expansion of  $A$  with an arbitrarily chosen linear order.*

This theorem is weaker than Gaifman's Theorem (Theorem 4.3.3) in that it does not provide us with a normal form which would allow us to reduce evaluating a sentence in a structure to evaluating basic local sentences, which is an important step in many model checking algorithms for first-order logic (cf. Chapter 6).

The kind of locality provided by Corollary 4.3.4 and Theorem 4.3.5, i.e. comparing the isomorphism types of neighbourhoods of tuples in a single structure, is sometimes referred to as *Gaifman locality* (cf. [HLN99]). The more stronger locality property provided by Theorem 4.3.3 allows one to relate the truth value of FO sentences in *different* structures based on their neighbourhood types; this is formalised in the concept of *Hanf locality* in [HLN99].

In [AvMSS12], Anderson et al. prove that  $\text{Arb-inv-FO}$  has Gaifman locality, but with a polylogarithmic (rather than constant) locality radius. Again,

we define Arb-inv-FO as a logic containing only *sentences*, but this definition can be adapted to formulae with free variables in the obvious way.

**Theorem 4.3.6.** *For every Arb-invariant first-order formula  $\varphi(x_1, \dots, x_k)$  there is function  $r : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ , such that  $r(n) \in O(\log^c n)$  for some constant  $c$  and the following holds: If  $A$  is a structure with  $|V(A)| = n$  and  $\bar{a}, \bar{b} \in V(A)^k$  are tuples such that*

$$N_A^{r(n)}(\bar{a}) \simeq N_A^{r(n)}(\bar{b}),$$

*then*

$$A' \models \varphi[\bar{a}] \quad \text{if, and only if,} \quad A' \models \varphi[\bar{b}],$$

*where  $A'$  is an expansion of  $A$  with the appropriate relations used invariantly by  $\varphi$ .*

*Furthermore, this locality radius is optimal.*

The proof of this theorem is based on the fact that FO formulae can be translated into families of circuits of polynomial size and depth bounded by the quantifier rank of the formula, i.e. AC<sub>0</sub> circuit families (cf. [Str94, BIS90]). Enriching FO with arbitrary invariantly used relations corresponds to fixing some of the inputs of these circuits, resulting in non-uniform circuit families if the used relations are not computable. Håstad's very strong lower bounds for these circuits (cf. [Hå86]) can then be used to prove locality of Arb-inv-FO.

## 4.4 Depth-Bounded Structures

While Benedikt and Segoufin proved that order-invariant FO has the same expressive power as plain FO on unsiblinged trees, their proof gives no information concerning the relative succinctness, i.e. about a potentially necessary blow-up in size, quantifier rank, or quantifier alternation depth, when passing from an <-inv-FO-sentence  $\varphi$  to an FO-sentence  $\psi$ .

We show a quantitative collapse result on structures of bounded tree-depth, based on Benedikt and Segoufin's result and the fact that tree decompositions of bounded depth can be defined not only in MSO but even in FO, a result that may be of independent interest. As a by-product of our quantitative analysis we find that the *quantifier alternation hierarchy*

for first-order logic collapses on structures of bounded treedepth: By a result of Chandra and Harel [CH82], for every  $k \geq 1$  there is an FO-formula  $\varphi_k \in \text{FO}$  of quantifier alternation depth  $k$  for which there is no equivalent FO-formula of quantifier alternation depth  $k - 1$ . In contrast to this every  $<\text{-inv-FO}$ -sentence is equivalent to an FO-sentence of quantifier alternation depth at most  $3d$  on all structures of treedepth at most  $d$ .

To be precise, we prove the following theorem:

**Theorem 4.4.1.** *For every  $d \geq 1$ , every signature  $\sigma$ , and each sentence  $\varphi$  of  $<\text{-inv-FO}$ , there is an  $\text{FO}(\sigma)$ -sentence  $\psi$  which is equivalent to  $\varphi$  on  $\sigma$ -structures of treedepth at most  $d$ , has size at most  $d$ -fold exponential in the quantifier rank of  $\varphi$ , and quantifier alternation depth at most  $3d$ .*

To prove this theorem we show that for every quantifier rank  $q$  and every structure  $A \in \text{TD}_d(\sigma)$  there exists a class of canonical linear orders  $\preceq_q$  for which the  $\text{FO}_q$ -type of  $(A, \preceq_q)$  is already FO-definable in  $A$ . In particular,  $\text{tp}_q(A, \preceq_q)$  only depends on  $A$ , even though there may be more than one such order on  $A$ .

We call these canonical orders  $q$ -orders. After defining them formally we will thus prove the following two facts about them:

1. Expansions by  $q$ -orders are indistinguishable in  $\text{FO}_q$ , i.e.

$$(A, \preceq_1) \equiv_q (A, \preceq_2)$$

for all finite structures  $A$ , provided both  $\preceq_1$  and  $\preceq_2$  are  $q$ -orders (this is proved in Lemma 4.4.5).

2. If the treedepth of structures is bounded, then the  $q$ -type  $\text{tp}_q(A, \preceq_q)$  of an expansion of  $A$  by a  $q$ -order is definable in FO (Lemmas 4.4.8 and 4.4.11). The proof of Theorem 4.4.1 easily follows from this.

**Encoding information about elements in extended signatures** In our proofs we will repeatedly remove single elements  $r$  from structures  $A$  and encode information about the relations between  $r$  and the remaining elements into an expansion  $A^{[r]}$  of the structure  $A \setminus r$  (which is the substructure of  $A$  induced on the elements different from  $r$ ). We do this in such a way that the  $q$ -type of  $A$  is determined by the  $q$ -type of  $A^{[r]}$  together with what we call the atomic type of  $r$  in  $A$ .

The *atomic type*  $\alpha(A, a)$  of an element  $a$  of a  $\sigma$ -structure  $A$  is the set of all  $R \in \sigma$  such that  $(a, \dots, a) \in R^A$  (where the tuple  $(a, \dots, a)$  has length



$\text{ar}(R)$ ). If no confusion seems likely we omit  $A$  and just write  $\alpha(a)$ . Thus an atomic type is a subset of  $\sigma$ , and we identify  $\alpha \subseteq \sigma$  with the  $\text{FO}(\sigma)$ -sentence

$$\alpha(x) := \bigwedge_{R \in \alpha} R\bar{x} \wedge \bigwedge_{R \in \sigma \setminus \alpha} \neg R\bar{x},$$

where  $\bar{x} = (x, \dots, x)$  is a tuple of appropriate length. Since we will often need the atomic type of the  $\leq$ -minimal element  $r$  of an ordered structure we denote this by  $\alpha_A := \alpha(r, A)$ .

To encode the relations between the element which is removed and the remaining elements, we define a signature  $\tilde{\sigma}$  which contains, for each  $R \in \sigma$  and each nonempty  $I \subseteq [\text{ar}(R)]$ , a relation symbol  $R_I$  of arity  $|I|$ . Given a structure  $A = (A, (R^A)_{R \in \sigma})$  and an element  $r \in A$  we now obtain a  $\tilde{\sigma}$ -structure  $A^{[r]} = (A, (R_I^{A^{[r]}})_{R_I \in \tilde{\sigma}})$  by setting

$$R_I^{A^{[r]}} := \left\{ (a_i)_{i \in I} \mid (a_1, \dots, a_{\text{ar}(R)}) \in R^A \text{ and } a_i = r \text{ for } i \notin I \right\}.$$

Note that  $R^A = R_{[\text{ar}(R)]}^{A^{[r]}}$ , so up to a renaming of relation symbols,  $A^{[r]}$  is an expansion of  $A \setminus r$ .

The  $(L, q)$ -type of  $A$  is determined by  $\alpha(r)$  and the  $(L, q)$ -type of  $A^{[r]}$ :

**Lemma 4.4.2.** *Let  $L \in \{\text{FO}, \text{MSO}\}$  and  $q \in \mathbb{N}^+$ . Let  $A$  and  $B$  be structures,  $r \in A$  and  $s \in B$ . If*

$$\alpha(A, r) = \alpha(B, s) \quad \text{and} \quad \text{tp}_q^L(A^{[r]}) = \text{tp}_q^L(B^{[s]}),$$

*then also*

$$\text{tp}_q^L(A) = \text{tp}_q^L(B).$$

*Proof.* The same argument works for  $L = \text{FO}$  and  $L = \text{MSO}$ . Duplicator has a winning strategy  $\mathcal{S}$  in the  $q$ -round Ehrenfeucht-Fraïssé game for  $L$  on  $A^{[r]}$  and  $B^{[s]}$ . Note that the strategy  $\mathcal{S}$  is, in particular, a winning strategy on  $A \setminus r$  and  $B \setminus s$ , because  $A^{[r]}$  and  $B^{[s]}$  are expansions of these structures. Duplicator can win the  $q$ -round Ehrenfeucht-Fraïssé-game on  $A$  and  $B$  if she plays according to  $\mathcal{S}$  on  $A \setminus r$  and  $B \setminus s$ , and if she responds to  $r$  with  $s$  and vice versa.

We have to argue that this strategy preserves relations between the chosen elements. For relations not involving the removed elements  $r$  and  $s$ , this is true because  $\mathcal{S}$  is a winning strategy for the  $q$ -round game on  $A \setminus r$  and

$B \setminus s$ . Relations involving only the minimal elements are preserved because  $\alpha(A, r) = \alpha(B, s)$ . Relations involving the minimal elements and other elements are preserved, because they are encoded in the relations  $R_I$  of the extended signature  $\tilde{\sigma}$ , and these are preserved by  $\mathcal{S}$ .  $\square$

Our definitions are geared towards the following lemma:

**Lemma 4.4.3.** *Let  $L \in \{FO, FMod\}$ . For every  $L(\tilde{\sigma})$ -sentence  $\varphi$  there is an  $L(\sigma)$ -formula  $\mathcal{I}(\varphi)(z)$  of the same quantifier rank and quantifier alternation depth such that*

$$A \models \mathcal{I}(\varphi)(r) \quad \text{iff} \quad A^{[r]} \models \varphi,$$

for all  $\sigma$ -structures  $A$  and  $r \in A$ .

*Proof.* The proof uses a standard interpretation argument. It suffices to provide quantifier-free formulae with a parameter  $z$  which define the universe and the relations of  $A^{[r]}$  in  $A$ , provided that  $z$  is interpreted by the element  $r$ . The universe is defined by the formula  $x \neq z$ . Let  $R_I \in \tilde{\sigma}$ . If, for each  $i \leq \text{ar}(R)$ , we let

$$y_i := \begin{cases} x_j & \text{if } i = i_j \in I \\ z & \text{if } i \notin I \end{cases}$$

then  $Ry_1 \dots y_{\text{ar}(R)}$  is a formula with free variables  $z, x_1, \dots, x_{|I|}$  which defines  $R_I^{A^{[r]}}$  in  $(A, r)$ .  $\square$

**The definition of  $q$ -orders** We define the notion of  $q$ -orders more generally for logics  $L \in \{FO, MSO\}$  since we will use them again in Chapter 5 to prove similar collapse results for  $<\text{inv-MSO}$ . We fix arbitrary orders  $\preceq_{L,q}$  on the set of  $(L, q)$ -types over the signature  $\sigma^<$ , and  $\preceq_{\text{atomic}}$  on the set of atomic  $\sigma$ -types. For simplicity we write  $a \preceq_{\text{atomic}} b$  for  $\alpha(a) \preceq_{\text{atomic}} \alpha(b)$ .

To obtain a  $q$ -order  $\preceq$  on a connected structure  $A \in \mathbf{TD}_d(\sigma)$ , we pick a root  $r$  of  $A$  which has  $\preceq_{\text{atomic}}$ -minimal atomic type among all roots and for which the type of  $q$ -ordered expansions of  $A^{[r]}$  is  $\preceq_{L,q}$ -minimal among all  $\preceq_{\text{atomic}}$ -minimal roots. We place this  $r$  in front of the order  $\preceq$  and order the remaining elements according to a (recursively obtained)  $q$ -order on  $A^{[r]}$ . On structures with more than one component, we  $q$ -order the components individually and take the sum of their orders, following the  $\preceq_{L,q}$ -order of the components:

**Definition 4.4.4** ( $(L, q)$ -order). Recall that  $\text{root}(A)$  is the set of treedepth roots of  $A$ , i.e. those vertices that may be at the roof of a minimum-height depth-first search tree (cf. Sec. 3.2). An  $(L, q)$ -order on a  $\sigma$ -structure  $A$  is an order  $\preceq$  which satisfies the following conditions:

- (1) If  $A$  is connected we denote by  $r \in A$  its  $\preceq$ -minimal element. Then either  $|A| = 1$ , or  $|A| > 1$  and the following holds:
  - (a)  $r$  is a  $\preceq_{\text{atomic}}$ -minimal root of  $A$ , i.e.  $r \in \text{root}(A)$  and  $r \preceq_{\text{atomic}} r'$  for all  $r' \in \text{root}(A)$ .
  - (b) The  $(L, q)$ -type of  $(L, q)$ -ordered expansions of  $A^{[r]}$  is minimal:

$$\text{tp}_q(A^{[r]}, \preceq) \preceq_{L, q} \text{tp}_q(A^{[r']}, \preceq')$$

for every  $r' \in \text{root}(A)$  with  $\alpha(r') = \alpha(r)$  and every  $(L, q)$ -order  $\preceq'$  on  $A^{[r']}$ .

- (c)  $\preceq|_{A \setminus r}$  is an  $(L, q)$ -order on  $A^{[r]}$ .
- (2) If  $A$  is not connected, we denote its components by  $A_1, \dots, A_\ell$  and set  $\preceq_i := \preceq|_{A_i}$ . Then  $\preceq$  is an  $(L, q)$ -order if
  - (a) each  $\preceq_i$  is an  $(L, q)$ -order of  $A_i$ , and
  - (b) after suitably permuting the components,

$$\preceq = \preceq_1 + \dots + \preceq_\ell \quad \text{and} \quad \text{tp}_q(A_i, \preceq_i) \preceq_{L, q} \text{tp}_q(A_j, \preceq_j) \text{ for } i \leq j.$$

We just speak of a  $q$ -order if the logic  $L$  is assumed to be clear from the context. The  $\preceq$ -minimal element of a  $q$ -order  $\preceq$  will be denoted by  $r_\preceq$ .

Definition 4.4.4 can be turned into an algorithm for constructing  $q$ -orders, showing that every structure can be  $q$ -ordered. Next we want to show that all  $q$ -ordered expansions  $(A, \preceq)$  of a given structure  $A$  have the same  $q$ -type, and that the  $q$ -type of  $(A^{[r_\preceq]}, \preceq)$  is also the same for all  $q$ -orders  $\preceq$  of  $A$ .

**Lemma 4.4.5.** *Let  $L \in \{FO, MSO\}$ ,  $q \in \mathbb{N}^+$ . For all  $(L, q)$ -orders  $\preceq, \preceq'$  of a structure  $A$ , we have*

$$(A, \preceq) \equiv_q^L (A, \preceq').$$

*If  $A$  is connected and  $\text{td}(A) > 1$ , then also  $(A^{[r_\preceq]}, \preceq) \equiv_q^L (A^{[r_{\preceq'}]}, \preceq')$ .*

*Proof.* The proof proceeds by induction on the size of  $A$ . If  $|A| = 1$  then  $\preceq = \preceq'$  and there is nothing to prove.

Let  $|A| > 1$  and suppose first that  $A$  is connected. By Definition 4.4.4,  $\alpha(r_{\preceq}) = \alpha(r_{\preceq'})$  and

$$\text{tp}_q(A^{[r_{\preceq}]}, \preceq) \preceq_{\mathbb{L}, q} \text{tp}_q(A^{[r_{\preceq'}]}, \preceq').$$

By symmetry also

$$\text{tp}_q(A^{[r_{\preceq'}]}, \preceq') \preceq_{\mathbb{L}, q} \text{tp}_q(A^{[r_{\preceq}]}, \preceq),$$

so  $\text{tp}_q(A^{[r_{\preceq}]}, \preceq) = \text{tp}_q(A^{[r_{\preceq'}]}, \preceq')$  and, by Lemma 4.4.2,  $(A, \preceq) \equiv_q (A, \preceq')$ .

Now consider the case where  $A$  is not connected, and let  $K_1, \dots, K_\ell$  be the components of  $A$ . By the definition of  $q$ -orders each  $K_i$  is  $q$ -ordered, so

$$(K_i, \preceq|_{K_i}) \equiv_q^{\mathbb{L}} (K_i, \preceq'|_{K_i})$$

for  $i = 1, \dots, \ell$  by what we have just said. Considering the way that an  $(\mathbb{L}, q)$ -order orders the components of a structure according to their  $(\mathbb{L}, q)$ -types (Part 2 of Definition 4.4.4), we obtain that  $(A, \preceq) \equiv_q^{\mathbb{L}} (A, \preceq')$  by repeatedly applying the Composition Lemma.  $\square$

By Lemma 4.4.5 it makes sense to speak of the  $q$ -order type of an unordered structure  $A$  which we define as

$$\text{tp}_q^<(A) := \text{tp}_q(A, \preceq_q).$$

If  $A$  is connected and  $\text{td}(A) > 1$ , we furthermore define its  $q$ -order root type as

$$\text{rtp}_q^<(A) := \text{tp}_q(A^{[r_{\preceq_q}]}, \preceq_q).$$

In both cases  $\preceq_q$  is some  $q$ -order on  $A$  and well-definedness is guaranteed by the lemma. Note that both these types are  $\sigma^<$ -types. Similarly, the atomic type  $\alpha_A := \alpha(r_{\preceq})$  of the minimal element in a  $q$ -ordered expansion of  $A$  is well-defined.

We set

$$\begin{aligned} \mathcal{T}_{\mathbb{L}, \sigma, q, d} &:= \{\text{tp}_q^<(A) \mid A \in \text{TD}_d(\sigma)\}, \\ \mathcal{T}_{\mathbb{L}, \sigma, q, d}^{\text{conn}} &:= \{\text{tp}_q^<(A) \mid A \in \text{TD}_d^{\text{conn}}(\sigma)\}, \text{ and} \\ \mathcal{T}_{\mathbb{L}, \sigma, q} &:= \bigcup_{d \in \mathbb{N}^+} \mathcal{T}_{\mathbb{L}, \sigma, q, d}. \end{aligned}$$

We say that a sentence  $\varphi_\tau \in \mathbf{L}(\sigma)$  *defines* a type  $\tau \in \mathcal{T}_{\mathbf{L},\sigma,q,d}$  on  $\mathbf{TD}_d(\sigma)$  (and that  $\tau$  is  $\mathbf{L}$ -definable) if for each  $A \in \mathbf{TD}_d(\sigma)$ , we have

$$A \models \varphi_\tau \quad \text{iff} \quad \text{tp}_q^<(A) = \tau.$$

Note that the sentence  $\varphi_\tau$  must not contain the relation  $<$ .

By Lemma 4.4.2 the atomic type of  $r_{\preceq}$  and the  $q$ -type of  $A^{[r_{\preceq}]}$  determine the  $q$ -type of  $A$ , and  $\text{td}(A^{[r_{\preceq}]}) = \text{td}(A) - 1$ , for connected structures  $A$  and  $q$ -orders  $\preceq$ . Since the number of atomic  $\tilde{\sigma}$ -types is  $2^{|\tilde{\sigma}|}$ , we obtain the following bound on the size of  $\mathcal{T}_{\sigma,q,d}^{\text{conn}}$ :

**Corollary 4.4.6.** *Let  $q, d \in \mathbb{N}^+$ . Then  $|\mathcal{T}_{\sigma,q,d}^{\text{conn}}| \leq 2^{|\tilde{\sigma}|} \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|$ .*

### Handling connected structures

The proof of our main theorem is broken down into two steps. In the first step, we show how to lift the definability of  $q$ -types of  $q$ -ordered structures from structures of treedepth  $d - 1$  to connected structures of treedepth  $d$ .

Again we invoke Lemma 4.4.2 and Lemma 4.4.5 to show that  $q$ -order types can be broken down into atomic types of roots and  $q$ -order root types:

**Corollary 4.4.7.** *Let  $d > 1$  and let  $\tau \in \mathcal{T}_{\sigma,q,d}^{\text{conn}}$ . Let*

$$R_\tau := \{(\alpha_A, \text{rtp}_q^<(A)) \mid A \in \mathbf{TD}_d^{\text{conn}}(\sigma), \text{td}(A) > 1, \text{ and } \text{tp}_q^<(A) = \tau\}.$$

*Then for each  $B \in \mathbf{TD}_d^{\text{conn}}(\sigma)$ , we have  $\text{tp}_q^<(B) = \tau$  iff  $(\alpha_B, \text{rtp}_q^<(B)) \in R_\tau$ .*

*Proof.* The “only-if”-part of the claim is obvious. Regarding the “if”-part, if

$$(\alpha_B, \text{rtp}_q^<(B)) = (\alpha_A, \text{rtp}_q^<(A))$$

for some  $A$  with  $\text{tp}_q^<(A) = \tau$ , then Lemma 4.4.5 and the definitions of  $\text{tp}_q^<, \text{rtp}_q^<$  imply that  $\text{tp}_q^<(B) = \tau$ .  $\square$

**Lemma 4.4.8.** *Let  $q, d \in \mathbb{N}^+$  with  $d > 1$ . Let  $(L_1, L_2)$  be one of  $(\text{FO}, \text{FO})$  or  $(\text{MSO}, \text{FOmod})$ . If each  $(L_1, q)$ -type  $\theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}$  is  $L_2(\tilde{\sigma})$ -definable on  $\text{Fin}_{\tilde{\sigma},d-1}$  by a sentence  $\psi_{\theta,d-1}$ , then each  $(L_1, q)$ -type  $\tau \in \mathcal{T}_{\sigma,q,d}^{\text{conn}}$  is  $L_2(\sigma)$ -definable on  $\mathbf{TD}_d^{\text{conn}}(\sigma)$  by a sentence  $\varphi_{\tau,d}^{\text{conn}}$ . Moreover, defining*

$$\Psi := \{\psi_{\theta,d-1} \mid \theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}\} \quad \text{and} \quad \Phi := \{\varphi_{\tau,d}^{\text{conn}} \mid \tau \in \mathcal{T}_{\sigma,q,d}\},$$

we have  $\|\Phi\| \leq c \cdot \|\Psi\| \cdot |\mathcal{T}_{\bar{\sigma},q,d-1}|^2$  and  $\text{qad}(\Psi) \leq \text{qad}(\Phi) + 1$ , for a constant  $c$  depending only on  $\sigma, d$ .

*Proof.* In the following, all  $q$ -types are  $(L_1, (\sigma^<), q)$ -types. Let  $\tau \in \mathcal{T}_{\sigma,q,d}^{\text{conn}}$  and let  $R_\tau$  be as in Corollary 4.4.7. We show that, under the assumptions of our lemma, the class

$$\{A \in \text{TD}_d^{\text{conn}}(\sigma) \mid (\alpha_A, \text{rtp}_q^<(A)) \in R_\tau\}$$

is  $L_2(\sigma)$ -definable by a sentence  $\varphi_\tau$  on  $\text{TD}_d^{\text{conn}}(\sigma)$ . Taking care of connected structures of treedepth 1 (i.e. singleton structures) we set  $\varphi_{\tau,d}^{\text{conn}} := (\text{td}_{\leq 1} \wedge \hat{\varphi}_\tau) \vee (\text{td}_{> 1} \wedge \varphi_\tau)$ , where  $\hat{\varphi}_\tau$  defines  $\tau$  on singleton structures.

For each atomic  $\sigma$ -type  $\alpha \subseteq \sigma$ , the following FO-sentence  $\xi_\alpha$  expresses in a structure  $A \in \text{TD}_d^{\text{conn}}(\sigma)$  that  $\alpha_A = \alpha$ :

$$\xi_\alpha := (\exists x (\text{root}_d(x) \wedge \alpha(x))) \wedge \left( \forall x (\text{root}_d(x) \rightarrow \bigvee_{\alpha \preceq_{\text{atomic}} \alpha'} \alpha'(x)) \right).$$

For each type  $\theta \in \mathcal{T}_{\bar{\sigma},q,d-1}$  the following sentence is true in a  $\sigma$ -structure  $A$  if, and only if, there is a root  $r$  of atomic type  $\alpha$  for which  $A^{[r]}$  has type  $\theta$ , and  $\theta$  is  $\preceq_{L_1,q}$ -minimal among the types of  $A^{[s]}$  for roots  $s$  of atomic type  $\alpha$ :

$$\begin{aligned} \chi_{\alpha,\theta} &:= \forall x \left( (\text{root}_d(x) \wedge \alpha(x)) \rightarrow \bigvee_{\theta \preceq_{L_1,q} \theta'} \mathcal{I}(\psi_{\theta',d-1})(x) \right) \\ &\wedge \exists x (\text{root}_d(x) \wedge \alpha(x) \wedge \mathcal{I}(\psi_{\theta,d-1})(x)). \end{aligned}$$

Observe that  $\text{qad}(\chi_{\alpha,\theta}) \leq \text{qad}(\Psi) + 1$ .

Now we obtain the desired sentence by defining

$$\varphi_\tau := \bigvee_{(\alpha,\theta) \in R_\tau} (\xi_\alpha \wedge \chi_{\alpha,\theta})$$

and, for some constant  $c$  depending only on  $\sigma, d$ , we have

$$\begin{aligned} \|\xi_\alpha\| &\leq c, \\ \|\chi_{\alpha,\theta}\| &\leq c \cdot \|\Psi\| \cdot |\mathcal{T}_{\bar{\sigma},q,d-1}|, \\ |R_\tau| &\leq c \cdot |\mathcal{T}_{\bar{\sigma},q,d-1}|, \text{ and} \\ \|\varphi_\tau\| &\leq c \cdot \|\Psi\| \cdot |\mathcal{T}_{\bar{\sigma},q,d-1}|^2. \end{aligned}$$

The claims about  $\|\Phi\|$  and  $\text{qad}(\Phi)$  follow from the observations above.  $\square$

## Handling disconnected structures

We proceed with the preparations for the second step in the proof of our main theorem, where we lift the definability of  $q$ -order types from connected structures of treedepth  $\leq d$  to disconnected structures of treedepth  $\leq d$ .

To us, a *Boolean query* is an isomorphism-invariant map  $f : \mathbf{Fin} \rightarrow \{0, 1\}$ , where  $\mathbf{Fin}$  is the class of all finite structures (i.e. structures over arbitrary signatures). We will treat maps  $f : \mathbf{Fin}_\sigma \rightarrow \{0, 1\}$  as Boolean queries by assuming that  $f(A) = 0$  if  $A$  is not a  $\sigma$ -structure. The general definition for arbitrary signatures will be useful in Chapter 5. We are interested in two kinds of queries. As usual, we identify each sentence  $\varphi$  with a Boolean query such that  $\varphi(A) = 1$  iff  $A \models \varphi$ . Furthermore, we identify each  $q$ -order type  $\tau$  with a query such that  $\tau(A) = 1$  iff  $\text{tp}_q^<(A) = \tau$ . For each structure  $A$  and each Boolean query  $f$ , we let  $n_f(A)$  denote the number of components  $K$  of  $A$  such that  $f(K) = 1$ . For each ordered set  $Q := \{f_1, \dots, f_\ell\}$  of Boolean queries, we let  $\bar{n}_Q(A) := (n_{f_1}(A), \dots, n_{f_\ell}(A))$ . For natural numbers  $a, b, t \in \mathbb{N}^+$  we set

$$a \equiv_{\wedge t} b \quad \Leftrightarrow \quad (a = b \text{ or } a, b \geq t),$$

and we extend this relation to tuples  $\bar{a}$  and  $\bar{b}$  by saying  $\bar{a} \equiv_{\wedge t} \bar{b}$  if, and only if,  $a_i \equiv_{\wedge t} b_i$  for all components  $a_i$  and  $b_i$ .

We show that FO inherits its capability to count the types of components in  $q$ -ordered structures from its capability to distinguish linear orders of different lengths. The proof of the following lemma closely follows a step in the proof of [BS09, Thm. 5.5]. Observe that

$$n_{\mathcal{T}_{\sigma,q,d}^{\text{conn}}}(A) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q,d}^{\text{conn}}}(B) \quad \text{if, and only if,} \quad n_{\mathcal{T}_{\sigma,q}}(A) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(B)$$

for all  $A, B \in \text{TD}_d(\sigma)$ .

**Lemma 4.4.9.** *Let  $d \geq 1$ ,  $q \in \mathbb{N}^+$  and  $t := 2^q + 1$ . Then for all  $A, B \in \text{TD}_d(\sigma)$ ,*

$$n_{\mathcal{T}_{\sigma,q}}(A) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(B) \implies \text{tp}_q^<(A) = \text{tp}_q^<(B).$$

*Proof.* For each component  $K$  of  $A$ , we let  $\preceq^K$  be a  $q$ -order of  $K$ . By Part 2 of Definition 4.4.4, the  $q$ -orders on the components of  $A$  can be extended to a  $q$ -order  $\preceq^A$  on  $A$  such that  $\preceq^A|_K = \preceq^K$  for each component  $K$  of  $A$ . We proceed analogously to obtain a  $q$ -order  $\preceq^B$  on  $B$ . Let  $\mathcal{T}_{\sigma,q} =$

$\{\tau_1, \dots, \tau_\ell\}$ , where  $\ell := |\mathcal{T}_{\sigma,q}|$  and  $\tau_i \preceq_q \tau_j$  iff  $i < j$ . We consider words over the alphabet  $\mathcal{T}_{\sigma,q}$  as structures in the usual way, i.e. as ordered structures over a signature containing a unary relation symbol for each type. Consider the words  $w_A, w_B \in \mathcal{T}_{\sigma,q}^*$  obtained from  $(A, \preceq^A)$  and  $(B, \preceq^B)$  by contracting each component  $K$  to a single element that gets labelled by its  $q$ -type in the corresponding  $q$ -ordered structure. By this construction and by Part 2 of Definition 4.4.4, we know that

$$w_A = \tau_1^{n_{\tau_1}(A)} \dots \tau_\ell^{n_{\tau_\ell}(A)} \quad \text{and} \quad w_B = \tau_1^{n_{\tau_1}(B)} \dots \tau_\ell^{n_{\tau_\ell}(B)}.$$

Since  $n_{\mathcal{T}_{\sigma,q}}(A) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(B)$ , for each  $i \in [\ell]$ , we have either  $n_{\tau_i}(A) = n_{\tau_i}(B)$  or  $n_{\tau_i}(A), n_{\tau_i}(B) \geq t$ . A folklore result (cf. [Lib04, Chapter 3]) tells us that  $w_A \equiv_q^{\text{FO}} w_B$ , i.e. Duplicator has a winning strategy in the  $q$ -round Ehrenfeucht-Fraïssé-game on the two word structures.

We show that  $(A, \preceq^A) \equiv_q^{\text{FO}} (B, \preceq^B)$ . To this end, consider the following winning strategy for Duplicator in the  $q$ -round Ehrenfeucht-Fraïssé-game on  $(A, \preceq^A)$  and  $(B, \preceq^B)$ . She maintains a *virtual*  $q$ -round Ehrenfeucht-Fraïssé-game  $w_A$  on  $w_B$  between a *Virtual Spoiler* and a *Virtual Duplicator*. When, during the  $i$ -th round, Spoiler chooses an element  $v$  in some component  $K$  of, say,  $A$ , she lets the Virtual Spoiler play the corresponding position in  $w_A$  in the  $i$ -th round of the virtual game. The Virtual Duplicator answers in  $w_B$ . Duplicator chooses a component  $K'$  of  $B$  for its reply according to the Virtual Duplicator's answer in  $w_B$ . The winning strategy on  $w_A$  and  $w_B$  ensures that  $(K, \preceq^A) \equiv_q^{\text{FO}} (K', \preceq^B)$  and that all elements of  $K$  and  $K'$  have the same positions in  $\preceq^A$  and  $\preceq^B$  relative to the elements played in the previous rounds. Duplicator uses her winning strategy in the  $q$ -round game on the ordered components to determine the element of  $K'$  that she uses as her answer to  $v$ .  $\square$

For a tuple  $\bar{a}$  of natural numbers, denote by  $[\bar{a}]_{\wedge t}$  the tuple obtained from it by replacing all entries  $> t$  with  $t$ . Then the previous lemma implies that if  $\text{td}(A) \leq d$ , then  $[\bar{n}_{\mathcal{T}_{C,q,d}^{\text{conn}}}(A)]_{\wedge (2^q+1)}$  determines  $\text{tp}^<(A)$ . Hence we obtain the following corollary:

**Corollary 4.4.10.** *Let  $q, d \in \mathbb{N}^+$  and let  $t := 2^q + 1$ . For each  $\varphi \in \text{FO}(\sigma^<)$ , let*

$$R_\varphi := \{[\bar{n}_{\mathcal{T}_{C,q,d}^{\text{conn}}}(A)]_{\wedge t} \mid A \in \text{TD}_d(\sigma), \text{tp}_q^<(A) \models \varphi\}.$$



Then for each  $A \in \mathbf{TD}_d(\sigma)$ , we have

$$\mathrm{tp}_q^<(A) \models \varphi \quad \text{if, and only if, } [\bar{n}\mathcal{T}_{C,q,d}^{\mathrm{conn}}]_{\wedge t} \in R_\tau.$$

Furthermore,  $|\mathcal{T}_{\sigma,q,d}| \leq (t+1)^{|\mathcal{T}_{C,q,d}^{\mathrm{conn}}|}$ .

The following lemma will be used in conjunction with the previous corollary to lift the definability of  $q$ -types from connected to disconnected structures.

**Lemma 4.4.11.** *Let  $L \in \{FO, FO\mathrm{mod}\}$ . For all  $d, t \in \mathbb{N}^+$ , every set of  $L$ -sentences  $\Phi$ , and every set  $R \subseteq [0, t]^{|\Phi|}$ , there is an  $L$ -sentence  $\psi_R^\Phi$  such that for each structure  $A$  with  $\mathrm{td}(A) \leq d$ , we have*

$$A \models \psi_R^\Phi \iff [\bar{n}\Phi(A)]_{\wedge t} \in R.$$

Moreover,  $\|\psi_R^\Phi\| \leq c \cdot |\Phi| \cdot \|\Phi\| \cdot |R| \cdot t^2$  and  $\mathrm{qad}(\psi_R^\Phi) \leq \mathrm{qad}(\Phi) + 2$ , for a constant  $c$  which depends only on  $\sigma, d$ .

*Proof.* Let  $\Phi := \{\varphi_1, \dots, \varphi_\ell\}$ . Consider some  $i \in [\ell]$  and let  $\tilde{\varphi}_i(x) := \varphi_i|_{\mathrm{reach}_d(x,z)}$ .

Let  $n \in [t]$ . We define a formula  $\psi_i^n(\bar{x})$ , where  $\bar{x} := (x_1, \dots, x_n)$ , which states that  $x_1, \dots, x_n$  lie in distinct connected components, each of which satisfies  $\varphi_i$ :

$$\psi_i^n(\bar{x}) := \bigwedge_{j \in [n]} \tilde{\varphi}_i(x_j) \wedge \bigwedge_{j, k \in [n], j \neq k} \neg \mathrm{reach}_d(x_j, x_k).$$

Observe that  $\mathrm{qad}(\psi_i^n) \leq \mathrm{qad}(\Phi)$  (in particular, since  $\mathrm{reach}_d$  is an existential formula) and that  $\|\psi_i^n\| \leq cn^2\|\Phi\| \leq ct^2\|\Phi\|$ , for a constant  $c$  depending on  $\sigma, d$  only.

To obtain a formula which states that either the (pairwise disjoint) components of the  $x_1, \dots, x_n$  are the only components which satisfy  $\varphi_i$  or the number of such components is at least  $t$ , we let

$$\psi_i^{n,t}(\bar{x}) := \begin{cases} \forall y \neg \tilde{\varphi}_i(y) & \text{if } n = 0, \\ \psi_i^n(\bar{x}) \wedge \forall y (\tilde{\varphi}_i(y) \rightarrow \bigvee_{i \in [n]} \mathrm{reach}_d(y, x_i)) & \text{if } 0 < n < t \\ \psi_i^n(\bar{x}) & \text{if } n \geq t. \end{cases}$$

Note that  $\mathrm{qad}(\psi_i^{n,t}) \leq \mathrm{qad}(\Phi) + 1$  and  $\|\psi_i^{n,t}\| \leq c \cdot \|\psi_i^n\|$ , for some constant  $c$  depending on  $\sigma, d$  only. (Note that  $\|\psi_i^n\| \geq n$ , so the disjunction over  $i \in [n]$

is absorbed by that.) We obtain the desired sentence  $\psi_{R,t}^\Phi$  by setting

$$\psi_{R,t}^\Phi := \bigvee_{(n_1, \dots, n_\ell) \in R} \exists \bar{x}_i \bigwedge_{i \in [\ell]} \psi_i^{n_i, t}(\bar{x}_i),$$

where  $\bar{x}_i$  is a tuple of  $n_i$  variables. Note that

$$\begin{aligned} \|\psi_R^\Phi\| &\leq |R| \cdot |\Phi| \cdot \max_{i \in [\ell]} \|\psi_i^t\| \leq c \cdot |R| \cdot |\Phi| \cdot \|\Phi\| \cdot t^2, \\ \text{qad}(\psi_R^\Phi) &\leq \max_{i \in [\ell]} \text{qad}(\psi_i^{n_i, t}) + 1 \leq \text{qad}(\Phi) + 2. \end{aligned}$$

□

We can now prove the main theorem of this section:

*Proof of Theorem 4.4.1.* By induction on the treedepth  $d$ , we show that for each signature  $\sigma$  and each  $\text{FO}(\sigma^<)$ -sentence  $\varphi$  with  $\text{qr}(\varphi) = q$ , there is an  $\text{FO}(\sigma)$ -sentence  $\psi_{\varphi, d}$  with  $\|\psi_{\varphi, d}\| \in d\text{-EXP}(q)$  and  $\text{qad}(\psi_{\varphi, d}) \leq 3d$  such that for each  $A \in \text{TD}_d(\sigma)$ , we have  $A \models \psi_{\varphi, d}$  iff  $\text{tp}_q^<(A) \models \varphi$ . Furthermore, we show that  $|\mathcal{T}_{\sigma, q, d}| \in d\text{-EXP}(q)$  and  $|\mathcal{T}_{\sigma, q, d}^{\text{conn}}| \in (d-1)\text{-EXP}(q)$ . To finish the proof, if  $\varphi$  is order-invariant, we let  $\psi := \psi_{\varphi, d}$ , and we obtain that  $A \models_{<} \varphi$  iff  $A \models \psi$ .

Let  $\mathcal{T}_{\sigma, q, d}^{\text{conn}} = \{\theta_1, \dots, \theta_\ell\}$ . First, for each  $i \in [\ell]$ , we construct a sentence  $\varphi_i$  that defines  $\theta_i$  on  $\text{TD}_d^{\text{conn}}(\sigma)$ . If  $d = 1$ , observe that any *connected* structure  $A$  of type  $\theta_i \in \mathcal{T}_{\sigma, q, 1}^{\text{conn}}$  consists of a single element. The atomic  $\sigma$ -type  $\alpha$  of this element determines the  $q$ -type of the unique  $q$ -order on  $A$ . The  $\text{FO}$ -sentence  $\varphi_{\tau, 1}^{\text{conn}} := \exists x \alpha(x)$  hence defines  $\tau$  on  $\text{TD}_1^{\text{conn}}(\sigma)$ . We obviously have  $\|\varphi_{\tau, 1}^{\text{conn}}\| \leq c \cdot |\sigma|$ , for some absolute constant  $c$ , and  $|\mathcal{T}_{\sigma, q, d}^{\text{conn}}| \leq 2^{|\sigma|} \in (d-1)\text{-EXP}(q)$ .

If  $d > 1$ , we construct an  $\text{FO}$ -sentence  $\psi_{\theta, d-1}$  inductively for each  $q$ -type  $\theta \in \mathcal{T}_{\sigma, q, d-1}$ . Let  $\Psi := \{\psi_{\theta, d-1} \mid \theta \in \mathcal{T}_{\sigma, q, d-1}\}$ . By induction, we obtain  $\|\Psi\| \in (d-1)\text{-EXP}(q)$ , and  $\text{qad}(\Psi) \leq 3(d-1)$ , and we have  $|\mathcal{T}_{\sigma, q, d-1}| \in (d-1)\text{-EXP}(q)$ . We construct  $\varphi_i$  according to Lemma 4.4.8, i.e. we let  $\varphi_i := \varphi_{\theta_i, d}^{\text{conn}}$  for each  $i \leq \ell$ . Let  $\Phi := \{\varphi_1, \dots, \varphi_\ell\}$ . Then there is a constant  $c$  depending only on  $\sigma, d$ , such that

$$\begin{aligned} \|\Phi\| &\leq c \cdot \|\Psi\| \cdot |\mathcal{T}_{\sigma, q, d-1}|^2 \in (d-1)\text{-EXP}(q) \quad \text{and} \\ \text{qad}(\Phi) &\leq \text{qad}(\Psi) + 2 \leq 3(d-1) + 2. \end{aligned}$$

Now consider a sentence  $\varphi \in \text{FO}(\sigma^<)$ . Let  $R := R_\varphi$  be given by Corollary 4.4.10. We apply Lemma 4.4.11 with  $t := 2^q + 1$  to obtain a sentence

$\psi_{\varphi,d} := \psi_R^\Phi$ . To see that  $\psi_{\varphi,d}$  is defined correctly, consider some  $A \in \mathbf{TD}_d(\sigma)$ . Observe that for each  $i \in [\ell]$  and each component  $K$  of  $A$ , we have  $K \models \varphi_i$  iff  $\text{tp}_q^<(K) = \tau_i$ , and thus  $\bar{n}_\Phi(A) = \bar{n}_{\mathcal{T}_{\sigma,q,d}^{\text{conn}}}(A)$ . Then

$$\begin{aligned} A \models \psi_{\varphi,d} &\text{ iff } [\bar{n}_{\mathcal{T}_{\sigma,q,d}^{\text{conn}}}(A)]_{\wedge t} \in R && \text{(by Lemma 4.4.11} \\ &&& \text{and the previous observation)} \\ &\text{ iff } \text{tp}_q^<(A) \models \varphi. && \text{(by Corollary 4.4.10)} \end{aligned}$$

By Lemma 4.4.11, for some constant  $c$  depending only on  $\sigma, d$ , we have

$$\begin{aligned} \|\psi_{\varphi,d}\| &\leq c \cdot |\Phi| \cdot |R| \cdot t^2 \cdot \|\Phi\| \quad \text{and} \\ \text{qad}(\psi_{\varphi,d}) &\leq \text{qad}(\Phi) + 1 \leq 3d. \end{aligned}$$

Observe that  $|\Phi| = \ell = |\mathcal{T}_{C,q,d}^{\text{conn}}| \in (d-1)\text{-EXP}(q)$  by Corollary 4.4.6 and that  $|R| \leq t^\ell \in d\text{-EXP}(q)$ . Hence,  $\|\psi_{\varphi,d}\| \in d\text{-EXP}(q)$ . By Corollary 4.4.10, we also obtain  $|\mathcal{T}_{C,q,d}| \in d\text{-EXP}(q)$ .  $\square$

## Chapter 5

# Order-Invariant Monadic Second-Order Logic

### 5.1 Depth-Bounded Structures

Courcelle proved in [Cou96, Thm. 4.1] that classes of graphs definable by order-invariant MSO sentences are recognisable in a certain algebraic sense. Recognisable sets of graphs of *bounded treewidth* are conjectured in [Cou91, Conjecture 1] to be definable in MSO with modulo-counting (CMSO), which would imply that  $<-inv$ -MSO is equivalent to CMSO on these graphs. Note that it is well-known and easy to see that, regardless of the considered class of structures, for each sentence of modulo-counting MSO there is an equivalent  $<-inv$ -MSO-sentence. Hence, the difficult part is the construction of an CMSO-sentence for a given  $<-inv$ -MSO-sentence.

The equivalence of recognisability and definability in CMSO for graphs of bounded treewidth has been shown by Bojańczyk and Pilipczuk in [BP16] (cf. also [BP17]) by showing that one can, in a graph of treewidth  $k$ , define a coloured tree encoding a width- $k$  tree decomposition of it using MSO. While Theorem 5.1.1 (which predates Bojańczyk and Pilipczuk's) works only for the further restricted case of structures of bounded *tree-depth*, it gives a stronger collapse result, namely to *first-order* logic with modulo counting (FOmod). Note that although  $<-inv$ -FO is known to be equivalent to FO on trees (cf. [BS05]) and MSO is known to be equivalent to FO on structures of bounded treedepth (cf. [EGT16]) one can not just combine these results in a black-box fashion. Furthermore we again obtain a bound on the quantifier alternation depth of the resulting formulae in terms of treedepth alone.

**Theorem 5.1.1.** *For every  $d \in \mathbb{N}^+$  and every  $<\text{-inv-MSO}$ -sentence  $\varphi$  there is an  $\text{FOmod}$ -sentence  $\psi$  with  $\text{qad}(\psi) \leq 3d$  which is equivalent to  $\varphi$  on  $\text{TD}_d(\sigma)$ .*

Unlike in Section 4.4 we do not analyse the formula size, because it is known from [GS05] that (plain) MSO can define the length of orders non-elementarily more succinctly than FO.

For the proof of Theorem 5.1.1, we proceed similarly to the last section. Again we need to understand  $<\text{-inv-MSO}$ 's capabilities to count the number of components of a given  $q$ -type in  $q$ -ordered structures. However, this time we need to count not only up to some threshold, but also modulo some fixed divisor.

For  $n \in \mathbb{N}$  and  $p \in \mathbb{N}^+$ , we let  $[n]_{\text{mod } p}$  denote the remainder of the division of  $n$  by  $p$ , and  $\bar{n} := (n_1, \dots, n_\ell) \in \mathbb{N}^\ell$ , we let

$$[\bar{n}]_{\text{mod } p} := ([n_1]_{\text{mod } p}, \dots, [n_\ell]_{\text{mod } p}).$$

Similarly, we set  $m \equiv_{\text{mod } p} n$  if  $p$  divides  $m - n$ , and extend this notion to tuples  $\bar{m}$  and  $\bar{n}$  component-wise.

Below, we prove the following Lemma which shows that MSO inherits its component counting capabilities in  $q$ -ordered structures from its capabilities to distinguish orders of different lengths.

**Lemma 5.1.2.** *For each  $q \in \mathbb{N}^+$ , there is a  $p \in \mathbb{N}^+$  such that for all  $q$ -ordered structures  $(A, \preceq^A)$  and  $(B, \preceq^B)$ ,*

$$\begin{aligned} (\bar{n}_{\mathcal{T}_{\sigma,q}}(A) \equiv_{\text{mod } p} \bar{n}_{\mathcal{T}_{\sigma,q}}(B) \text{ and } \bar{n}_{\mathcal{T}_{\sigma,q}}(A) \equiv_{\wedge p} \bar{n}_{\mathcal{T}_{\sigma,q}}(B)) \\ \implies (A, \preceq^A) \equiv_q^{\text{MSO}} (B, \preceq^B). \end{aligned}$$

In the following, we say that an ordered structure  $(A, \preceq)$  is *component ordered*, if the order  $\preceq$  is a sum of the orders on the components of  $A$ , i.e. for some enumeration  $K_1, \dots, K_n$  of the components of  $A$ , we have

$$\preceq = \preceq|_{K_1} + \preceq|_{K_2} + \dots + \preceq|_{K_n}.$$

Observe that  $q$ -ordered structures are also component ordered. It will be convenient to have some notation that allows us to treat component ordered structures similarly to words. Given two ordered structures  $(A, \preceq^A)$  and  $(B, \preceq^B)$ , we let

$$(A, \preceq^A) \sqcup (B, \preceq^B) := (A \sqcup B, \preceq^A + \preceq^B),$$

where  $A \sqcup B$  denotes the disjoint union of  $A$  and  $B$  and we consider  $\preceq^A, \preceq^B$  as orders on the components of the disjoint union (via the inclusion mappings for  $A, B$ ). Instead of  $(A, \preceq^A) \sqcup (B, \preceq^B)$ , we also write  $(A, \preceq^A)(B, \preceq^B)$ . Like in the following definition, we often omit the order to make this notation less cluttered. For each component ordered structure  $A$ , we define its  $i$ -th power  $A^i$  by  $A^1 := A$  and  $A^i := A^{i-1}A$  if  $i > 1$ .

The proof of Lemma 5.1.2 rests on the following lemma.

**Lemma 5.1.3** (Pumping Lemma). *For each  $q \in \mathbb{N}^+$ , there is a number  $p \in \mathbb{N}^+$  such that for all component ordered structures  $A$  and all  $r \in \mathbb{N}$ ,  $i, j \in \mathbb{N}^+$ ,*

$$A^{r+ip} \equiv_q^{\text{MSO}} A^{r+jp}.$$

*Proof.* Let  $\mathcal{T}$  denote the (finite) set of  $q$ -types which are realised by component ordered  $\sigma$ -structures. We lift the disjoint union of ordered structures to  $\mathcal{T}$  by defining  $\text{tp}_q(A) \sqcup \text{tp}_q(B) := \text{tp}_q(A \sqcup B)$ . The Feferman-Vaught Theorem (Theorem 1.1.16) shows that this operation is well-defined. It is also associative, so that  $(\mathcal{T}, \sqcup)$  is a finite semigroup. Hence, there is a number  $p$  such that for each  $\tau \in \mathcal{T}$ ,  $\tau^p$  is idempotent (cf. e.g. [How76]), i.e.  $\tau^p = \tau^{ip}$  for each  $i \in \mathbb{N}^+$ . Then, for all  $A, r, i, p$  as in the statement of the lemma,  $\text{tp}_q(A)^{r+ip} = \text{tp}_q(A)^{r+jp}$ , i.e.  $A^{r+ip} \equiv_q^{\text{MSO}} B^{r+jp}$ .  $\square$

*Proof of Lemma 5.1.2.* Let  $\mathcal{T}_{\sigma,q} = \{\tau_1, \dots, \tau_\ell\}$  with  $\tau_i \prec_q \tau_j$  iff  $i < j$ . For each  $i \in [\ell]$ , fix a connected  $q$ -ordered structure  $K_i$  whose type is  $\text{tp}_q(K_i) = \tau_i$ . By repeated application of the Feferman-Vaught Theorem, we can assume without loss of generality that  $K \simeq K_i$  for each  $q$ -ordered component  $K$  of  $A$  or  $B$  with  $\text{tp}_q(K) = \tau_i$ . Let  $n_i := n_{\tau_i}(A)$  and let  $m_i := n_{\tau_i}(B)$  for each  $i \in [\ell]$ . By part 2 of Definition 4.4.4, we obtain

$$A \simeq K_1^{n_1} K_2^{n_2} \dots K_\ell^{n_\ell} \quad \text{and} \quad B \simeq K_1^{m_1} K_2^{m_2} \dots K_\ell^{m_\ell}.$$

For each  $i \in [\ell]$ , we have  $n_{\tau_i}(A) \equiv_{\text{mod } p} n_{\tau_i}(B)$ , i.e. there are  $r_i \in [0, p-1]$  and  $a_i, b_i \in \mathbb{N}$  such that  $n_i = r_i + a_i p$  and  $m_i = r_i + b_i p$ . Furthermore, as  $n_{\tau_i}(A) \equiv_{\wedge p} n_{\tau_i}(B)$ , we have  $a_i > 0$  iff  $b_i > 0$ . By repeated application of the Pumping Lemma, we obtain

$$K_1^{n_1} K_2^{n_2} \dots K_\ell^{n_\ell} \equiv_q^{\text{MSO}} K_1^{r_1+b_1 p} K_2^{r_2+b_2 p} \dots K_\ell^{r_\ell+b_\ell p} = K_1^{m_1} K_2^{m_2} \dots K_\ell^{m_\ell}.$$

Hence,  $A \equiv_q^{\text{MSO}} B$ .  $\square$

The next lemma is a modulo-counting analogue of Lemma 4.4.11.

**Lemma 5.1.4.** *For all  $d, p \in \mathbb{N}^+$ , each set of  $\text{FOMod}(\sigma)$ -sentences  $\Phi$ , and each set  $R \subseteq [0, p]^\ell \times [0, p-1]^\ell$  (where  $\ell = |\Phi|$ ), there is an  $\text{FOMod}(\sigma)$ -sentence  $\chi_R^\Phi$  such that for each  $A \in \text{TD}_d(\sigma)$ ,*

$$A \models \chi_R^\Phi \quad \text{iff} \quad ([\bar{n}_\Phi(A)]_{\wedge p}, [\bar{n}_\Phi(A)]_{\text{mod } p}) \in R.$$

Furthermore,  $\text{qad}(\chi_R^\Phi) \leq \max\{\text{qad}(\Phi) + 2, 2(d-1) + 1\}$ .

In contrast to Lemma 4.4.11, the proof of Lemma 5.1.4 is not straightforward, because it is not obvious how modulo-counting quantifiers can be used to count the number of components satisfying a given  $\text{FOMod}$ -sentence. A remedy to this problem is provided by Lemma 3.2.1, which shows that the number of tree-depth roots of each component of a graph (and hence of a structure) can be bounded in terms of its tree-depth only.

*Proof of Lemma 5.1.4.* Let  $\Phi = \{\varphi_1, \dots, \varphi_\ell\}$ . For each  $\bar{n} \in [0, p]^\ell$ , let  $\varphi_{\{\bar{n}\}}^\Phi$  be given by Lemma 4.4.11 for  $t := p$ , i.e. for each  $A \in \text{TD}_d(\sigma)$ , we have  $A \models \varphi_{\{\bar{n}\}}^\Phi$  iff  $[\bar{n}_\Phi(A)]_{\wedge p} = \bar{n}$ . Furthermore,  $\text{qad}(\varphi_{\{\bar{n}\}}^\Phi) \leq \text{qad}(\Phi) + 2$ . Below, for each  $\bar{r} := (r_1, \dots, r_\ell) \in [0, p-1]^\ell$ ,  $i \in [\ell]$ , we construct a sentence  $\chi_i^{\bar{r}}$  such that  $A \models \chi_i^{\bar{r}}$  iff  $n_{\varphi_i}(A) \equiv_{\text{mod } p} r_i$ . Furthermore,  $\text{qad}(\chi_i^{\bar{r}}) \leq \max\{\text{qad}(\Phi) + 1, 2(d-1) + 2\}$ . We can then define

$$\chi_R^\Phi := \bigvee_{(\bar{n}, \bar{r}) \in R} (\varphi_{\{\bar{n}\}}^\Phi \wedge \bigwedge_{i \in [\ell]} \chi_i^{\bar{r}}).$$

Obviously,  $\text{qad}(\chi_R^\Phi) \leq \max\{\text{qad}(\Phi) + 2, 2(d-1) + 2\}$ .

Consider some  $\bar{r} := (r_1, \dots, r_\ell) \in [0, p-1]^\ell$ ,  $i \in [\ell]$ , and let  $\varphi := \varphi_i$  and  $r := r_i$ . We define a formula  $\varphi^{=k}(x)$ , such that  $A \models \varphi^{=k}(a)$ , for  $A \in \text{TD}_d(\sigma)$  and  $a \in A$ , iff  $a$  belongs to a component  $K$  of  $A$  such that  $K \models \varphi$ ,  $a \in \text{root}(K)$ , and  $|\text{root}(K)| = k$ . Let  $\tilde{\varphi}(x) := \varphi|_{\text{reach}_d(x, z)}$ , let  $\tilde{\text{root}}_d(x) := \text{root}_d(x)|_{\text{reach}_d(x, z)}(x)$ , and let

$$\begin{aligned} \varphi^{=k}(x) &:= \tilde{\varphi}(x) \wedge \tilde{\text{root}}_d(x) \\ &\wedge \exists x_1 \dots \exists x_k \left( \bigwedge_{j \in [k]} (\tilde{\text{root}}_d(x_j) \wedge \text{reach}_d(x_j, x)) \wedge \bigwedge_{j, j' \in [k], j \neq j'} x_j \neq x_{j'} \right) \\ &\wedge \forall y \left( \tilde{\text{root}}_d(y) \wedge \bigwedge_{j \in [k]} y \neq x_j \rightarrow \bigwedge_{j \in [k]} \neg \text{reach}_d(y, x_j) \right). \end{aligned}$$

Observe that

$$\begin{aligned} \text{qad}(\varphi^{=k}) &\leq \max\{\text{qad}(\tilde{\varphi}), \text{qad}(\tilde{\text{root}}_d) + 1, \text{qad}(\text{reach}_d) + 1\} \\ &\leq \max\{\text{qad}(\varphi), 2(d-1) + 1\}. \end{aligned}$$

Let the function  $f$  be defined as in Lemma 3.2.1 and let  $b := f(d)$ . Let  $M \subseteq [0, p-1]^{b+1}$  be such that

$$(a_0, \dots, a_b) \in M \quad \text{iff} \quad \sum_{k \in [0, b]} k \cdot a_k \equiv_{\text{mod } p} r.$$

Now we define our formula  $\chi_i^{\bar{n}}$  as

$$\chi_i^{\bar{n}} := \bigvee_{(a_1, \dots, a_b) \in M} \bigwedge_{k \in [0, b]} \exists^{k \cdot a_k \pmod{p}} x \varphi^{=k}(x).$$

Obviously,  $\text{qad}(\chi_i^{\bar{n}}) \leq \max\{\text{qad}(\varphi), 2(d-1) + 1\} + 1$ .

We show that the formula is defined correctly. Let  $A \in \text{TD}_d(\sigma)$ . Recall that, according to Lemma 3.2.1,  $|\text{root}(K)| \leq b$  for each component  $K$  of  $A$ . We partition the set  $H$  of components of  $A$  into pairwise disjoint sets  $H_0, \dots, H_b$  such that  $K \in H_k$  iff  $|\text{root}(K)| = k$ , for each  $K \in H$ . By definition of  $\varphi^{=k}(x)$ , the number of elements  $a \in A$  such that  $A \models \varphi^{=k}(a)$  equals  $k \cdot |H_k|$ . Hence,  $A \models \chi_i^{\bar{n}}$  iff for some  $(a_0, \dots, a_b) \in M$ , we have  $k \cdot |H_k| \equiv k \cdot a_k \pmod{p}$  for each  $k \in [0, b]$ . This is true iff  $n_\varphi(A) \equiv r \pmod{p}$ , since

$$n_\varphi(A) = \sum_{k \in [0, b]} k \cdot |H_k| \equiv_{\text{mod } p} \sum_{k \in [0, b]} k \cdot a_k \equiv_{\text{mod } p} r,$$

for  $a_0, \dots, a_b \in [0, p-1]$  such that  $|H_k| \equiv_{\text{mod } p} a_k$  for each  $k \in [0, b]$ .  $\square$

With these preparations, the proof of Theorem 5.1.1 is very similar to the proof of Theorem 4.4.1.

*Proof of Theorem 5.1.1.* The proof proceeds by induction on the tree-depth  $d$ . We show that for each  $\text{MSO}(\sigma^<)$ -sentence  $\varphi$  with  $\text{qr}(\varphi) = q$ , there is an  $\text{FOmod}(\sigma)$ -sentence  $\psi_{\varphi, d}$  such that for each  $A \in \text{TD}_d(\sigma)$ , we have  $A \models \psi_{\varphi, d}$  iff  $\text{tp}_q^<(A) \models \varphi$ . In particular, if  $\varphi$  is order-invariant, we let  $\psi := \psi_{\varphi, d}$ , and we obtain  $A \models_{<} \varphi$  iff  $A \models \psi := \psi_{\varphi, d}$ .

Let  $\mathcal{T}_{\sigma, q, d}^{\text{conn}} = \{\theta_1, \dots, \theta_\ell\}$ . We construct a sentence  $\varphi_i$  that defines  $\theta_i$  on  $\text{TD}_d^{\text{conn}}(\sigma)$ , for each  $i \in [\ell]$ . If  $d = 1$ , the type of a connected structure of



type  $\theta_i$  is determined by the atomic  $\sigma$ -type  $\alpha$  of its single element. We let  $\varphi_{\tau,1}^{\text{conn}} := \exists x \alpha(x)$ . If  $d > 1$ , for each  $q$ -type  $\theta \in \mathcal{T}_{\bar{\sigma},q,d-1}$ , we obtain an **FOmod**-sentence  $\psi_{\theta,d-1}$  with  $\text{qad}(\psi_{\theta,d-1}) \leq 3(d-1)$ .

We construct  $\varphi_i$  according to Lemma 4.4.8, i.e. we let  $\varphi_i := \psi_{\theta_i,d}^{\text{conn}}$  for each  $i \leq \ell$ . Let  $\Phi := \{\varphi_1, \dots, \varphi_\ell\}$ . Note that  $\text{qad}(\Phi) \leq 3(d-1) + 2$ .

Now consider a sentence  $\varphi \in \text{MSO}(\sigma^<)$ . Let

$$R := \left\{ ([\bar{n}_{\mathcal{T}_{\sigma,q}}(B)]_{\wedge p}, [\bar{n}_{\mathcal{T}_{\sigma,q}}(B)]_{\text{mod } p}) \mid B \in \text{TD}_d(\sigma), \text{tp}_q^<(B) \models \varphi \right\}$$

where  $p$  is given by the Pumping Lemma for  $q$ . We construct  $\psi_{\varphi,d} := \psi_R^\Phi$  according to Lemma 5.1.4. In particular,  $\text{qad}(\psi_{\varphi,d}) \leq \text{qad}(\Phi) + 1 \leq 3d$ . Consider some  $A \in \text{TD}_d(\sigma)$ . Observe that, for each component  $K$  of  $A$ , we have  $K \models \varphi_i$  iff  $\text{tp}_q^<(K) = \tau_i$ . Hence,

$$([\bar{n}_\Phi(A)]_{\wedge p}, [\bar{n}_\Phi(A)]_{\text{mod } p}) = ([\bar{n}_{\mathcal{T}_{\sigma,q}}(A)]_{\wedge p}, [\bar{n}_{\mathcal{T}_{\sigma,q}}(A)]_{\text{mod } p}),$$

and thus

$$\begin{aligned} A \models \psi_{\varphi,d} &\iff \\ ([\bar{n}_{\mathcal{T}_{\sigma,q}}(A)]_{\wedge p}, [\bar{n}_{\mathcal{T}_{\sigma,q}}(A)]_{\text{mod } p}) &= ([\bar{n}_{\mathcal{T}_{\sigma,q}}(B)]_{\wedge p}, [\bar{n}_{\mathcal{T}_{\sigma,q}}(B)]_{\text{mod } p}) \end{aligned}$$

for some structure  $B \in \text{TD}_d(\sigma)$  with  $\text{tp}_q^<(B) \models \varphi$ . As a consequence of Lemma 5.1.2, this holds iff  $\text{tp}_q^<(A) \models \varphi$ .  $\square$

## 5.2 Decomposable Structures

Already on sets,  $<$ -inv-MSO is stronger than MSO because of its ability to do modulo counting: A linearly ordered set has even cardinality if, and only if, it has a subset containing the first but not the last element and exactly one of each pair of consecutive elements. The yardstick to compare  $<$ -inv-MSO to is therefore not MSO but its counting variant CMSO.

In general,  $<$ -inv-MSO is strictly stronger than CMSO, as proved by Ganzow and Rubin [GR08]. To separate these two logics, Ganzow and Rubin defined a query of grid-like structures and showed that it is definable in  $<$ -inv-MSO but not in CMSO. As grids are known to have large treewidth, this does not rule out the possibility that  $<$ -inv-MSO  $\equiv$  CMSO on graphs of bounded treewidth.

In fact, Courcelle showed in [Cou96] that  $<$ -inv-MSO  $\equiv$  CMSO on trees, and conjectured that this is true for every class of structures of bounded

treewidth. This was proved by Bojanczyk and Pilipczuk in 2016 [BP16] by showing that tree decompositions of bounded width are definable in MSO.

Using different techniques, Elberfeld et al. showed in [EFG16] that  $<-inv\text{-MSO} \equiv \text{CMSO}$  on

- classes of structures of bounded treewidth and
- classes of structures excluding the complete bipartite graph  $K_{3,\ell}$  for some  $\ell \in \mathbb{N}$  as a minor. This includes planar graphs, since these do not contain  $K_{3,3}$  as a minor by Kuratowski's Theorem [Die12, Section 4.4].



## Part III

# Algorithmic Meta-Theorems



## Chapter 6

# The Complexity of Model Checking

The model checking problem for a logic  $L$  on a class  $\mathcal{C}$  of (finite) structures is the following computational problem: Given a sentence  $\varphi \in L(\sigma)$  and a finite  $\sigma$ -structure  $A$ , both suitably encoded as a string, decide whether  $A \models \varphi$ . Variants of this problem ask for variable bindings or for the number of variable bindings which satisfy a given formula in a structure, but we restrict ourselves to the decision variant here.

We will not concern ourselves with details of the encoding, neither of the structure  $A$  nor of the sentence  $\varphi$ . Essentially two natural encoding schemes for structures present themselves, namely encoding relations by listing the tuples they contain, or in some kind of matrix. We assume the former type, resulting in strings of length approximately

$$O\left((\log |V(A)|) \cdot \left(\sum_{R \in \sigma} |R^A| \cdot \text{ar}(R)\right)\right)$$

for relational signatures, and similar for arbitrary signatures. As for the computational model, we assume random access to the input structure and the ability to perform basic arithmetic on  $(\log |V(A)|)$ -bit numbers in a single step.

Traditionally, the running time of algorithms is measured as the maximum number of computation steps needed by the algorithm as a function of the length of the input, cf. [Pap03, AB09], for example. A decision problem is then considered tractable if there is some algorithm solving it in polynomial running time.

The input to the model checking problem consists of two parts, the structure  $A$  and the formula  $\varphi$ . If both are treated equally, the model checking problem is not likely to be tractable even for first-order logic when  $A$  is restricted to be a two-element structure. This is because the PSPACE-complete problem of deciding whether a *quantified Boolean formula* evaluates to true or not (again cf. [Pap03, AB09]) is easily reducible to this problem.

It is therefore natural to treat the two parts of the input differently. Since it is reasonable to assume the formula to be much smaller than the structure  $A$ , we investigate the *parameterised complexity* (cf. [DF99, FG06]) of the model checking problem, with the formula  $\varphi$  or the length  $|\varphi|$  of its encoding as the parameter. The most common notion of tractability is then that of *fixed-parameter tractability* (*fpt*), i.e. the existence of an algorithm with running time bounded by

$$f(\varphi) \cdot |V(A)|^c$$

for some computable function  $f$  and some constant  $c \in \mathbb{N}$ . Note that the constant  $c$  must be independent of the formula  $\varphi$ .

Even in this relaxed sense, model checking for first-order logic on arbitrary finite structures is not likely to be tractable, as it has been shown to be complete for the parameterised complexity class  $\text{AW}[*]$  (cf. [FG06, Sec. 8.6]). While model checking for first-order logic is readily seen to be in PTIME for every fixed formula  $\varphi$  (placing the problem inside a parameterised complexity class called uniform XP), for monadic second-order logic even this is not the case; in fact for a fixed MSO-formula model checking may be complete for any level of the polynomial hierarchy.

Thus in order to obtain fixed-parameter tractable variants of the model checking problem one has to further restrict the problem. A common way of doing this is by only allowing structures  $A$  from certain classes of finite structures as input. One of the first such results was found by Courcelle:

**Theorem 6.0.1** (Courcelle [Cou90]). *There is a computable function  $f$  and an algorithm which decides for every structure  $A$  whose Gaifman graph has treewidth at most  $k$  and every  $\varphi \in \text{MSO}$  whether*

$$A \models \varphi$$

*in time  $f(\varphi, k) \cdot |A|$ .*

This hinges on an algorithm due to Bodlaender [Bod96] which, given a graph  $G = (V, E)$ , computes a tree decomposition of width  $k$ , if one exists, in time  $g(k) \cdot |V|$  for some computable function  $g$ .

**FO Model Checking** For first-order logic, model checking has been shown to be fixed-parameter tractable on a variety of classes, as depicted in Figure 6.1.

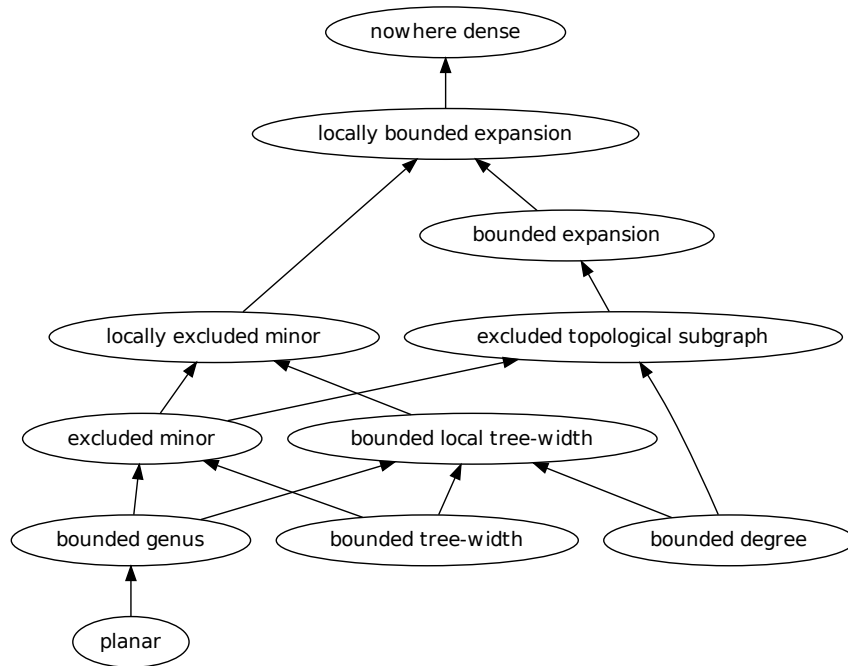


Figure 6.1: An overview of various classes of structures on which model checking for first-order logic has been shown to be fixed-parameter tractable. Arrows indicate inclusion relations between classes.

For structures whose Gaifman graph has bounded degree this was proved by Seese in [See96], for bounded treewidth it is a special case of Courcelle's Theorem (Thm. 6.0.1). For planar graphs this was shown by Frick and Grohe [FG01] using Gaifman's Theorem (Thm. 4.3.3) and the fact that planar graphs of bounded radius have bounded treewidth. Since graphs excluding some graph as a minor can be tree-decomposed into graphs with locally bounded treewidth [Gro03], one obtains the result for graphs with excluded



minors. For classes of graphs with bounded expansion, fixed-parameter tractability of FO-model checking was proved by Dvořák et al. in [DKT10] using low tree-depth colourings. Finally, Grohe et al. [GKS17] extended this to nowhere dense classes of graphs.

For monotone graph classes, i.e. classes of graphs which are closed under taking subgraphs, the result by Grohe et al. is optimal: Under standard complexity theoretic assumptions (namely  $\text{FPT} \neq \text{W}[1]$ ), there is no fixed-parameter tractable model checking algorithm for first-order logic on a monotone graph class that is not nowhere dense (cf. [Kre11] and [DKT10]).

**Vertex-Ordered Graphs** We show that, if we want to obtain efficient model checking algorithms for successor-invariant or order-invariant first-order logic, we do need to use the fact that we may arbitrarily choose the successor relation or linear order. In fact, there are classes of very simple graphs on which FO model checking is already  $\text{AW}[*]$ -complete if the graphs are equipped with a linear order, and slightly more complicated graphs for which this is true if the graphs are equipped with a successor relation. We presented these results in [EvdHK<sup>+</sup>19].

**Theorem 6.0.2.** *Let  $\mathcal{C}$  be a class of graphs. If  $\mathcal{C}$  contains all partial matchings, then model checking for first-order logic on the class of ordered expansions of graphs in  $\mathcal{C}$  is  $\text{AW}[*]$ -hard.*

*If  $\mathcal{C}$  contains all star forests, then model checking for first-order logic on the class of all successor-expansions of graphs in  $\mathcal{C}$  is  $\text{AW}[*]$ -hard.*

Here, a *partial matching* is a disjoint union of edges and isolated vertices (i.e. a graph of maximum degree 1), while a *star forest* is a disjoint union of stars (complete bipartite graphs  $K_{1,n}$  with  $n \geq 0$ ). Note that on both these graph classes, the model checking problem for plain FO is fixed-parameter tractable.

*Proof.* For the first part we show how to construct in polynomial time for every graph  $G$  an  $\{E, <\}$ -structure  $A$  with  $A|_{\{E\}} \in \mathcal{C}$  such that  $G$  can be FO-interpreted in  $A$ . For this, we let  $\{v_1, \dots, v_n\}$  be the vertex set of  $G$  ordered in an arbitrary way, and denote the degree of  $v_i$  in  $G$  by  $d_i$ . To each vertex in  $G$  we associate an interval of length  $\hat{d}_i := \max\{d_i, 1\}$  in  $A$ , and separate the intervals by gaps of length 2. Formally, setting

$$D_k := 2k - 1 + \sum_{i=1}^{k-1} \hat{d}_i \quad \text{for } k = 1, \dots, n,$$

we associate with  $v_i$  the interval  $\{D_i, \dots, D_i + \hat{d}_i - 1\}$ . The edge set  $E(A)$  consists of the edges  $\{D_i - 2, D_i - 1\}$  for  $i \geq 2$ , together with the edges  $\{D_i + k, D_j + \ell\}$  if  $v_i v_j$  is an edge of  $G$ ,  $v_j$  is the  $k$ -th neighbour of  $v_i$  in the ordering, and  $v_i$  is the  $\ell$ -th neighbour of  $v_i$ . Notice that the edges  $\{D_i - 2, D_i - 1\}$  are the only edges between consecutive elements, so they can be used to determine the intervals used in this construction.

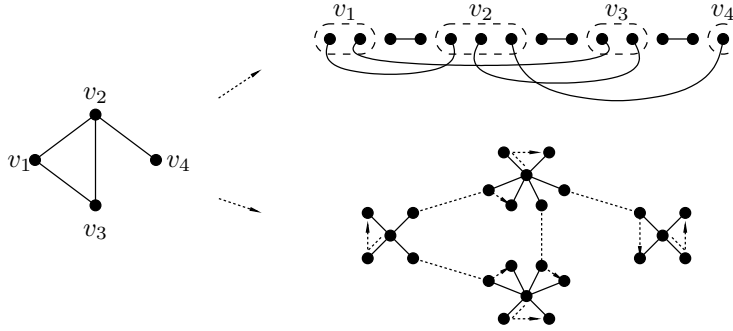


Figure 6.2: A sample graph (left) encoded in a linear order plus perfect matching (upper right) and a star forest plus successor relation (lower right).

For the second part we construct a structure  $A'$  consisting of a disjoint union of stars and a successor relation that can be used to recover the original graph using an FOinterpretation. Again, we assume the vertex set of  $G$  to be  $\{v_1, \dots, v_n\}$ . A vertex  $v$  is encoded by a path  $v^{-1}, v, v^{+1}$ . The vertices of these paths are placed at the beginning of the successor relation in an arbitrary order. An edge  $e$  is encoded by three vertices  $e^{-1}, e, e^{+1}$  such that  $e$  is a direct successor of  $e^{-1}$  and  $e^{+1}$  is a direct successor of  $e$ . All these vertices are placed at the end of the successor relation. For every edge  $e = vw$ , assume that  $v$  is smaller than  $w$  in the successor relation. We connect, in  $A'$ ,  $v$  to  $e^{-1}$  and  $w$  to  $e^{+1}$ . Again,  $G$  may be recovered from  $A'$  using an FOinterpretation.  $\square$

This theorem implies that FO-model checking is AW[\*]-hard on the classes of ordered expansions or successor expansions of many classes of graphs, such as planar graphs or forests. One notable exception is the class of graphs of bounded degree, on whose successor expansions FO-model checking is fixed-parameter tractable:

**Theorem 6.0.3.** *For every  $d \geq 0$  let  $\mathcal{D}_d$  be the class of graphs of maximum degree at most  $d$ . Then for all  $d \geq 0$ , model checking for FO is fixed-parameter tractable on the class of all successor expansions. In fact, we can*

*allow any (fixed) number of successor relations on top of  $\mathcal{D}_d$  and still have tractable first-order model-checking.*

*Proof.* By a result of Seese [See96] the model-checking problem for FO on graphs of bounded degree and also on all structures with Gaifman-graph of bounded degree is fixed-parameter tractable. Adding a successor relation increases the degree of the Gaifman-graph of a structure by at most two.  $\square$

## Chapter 7

# Successor-Invariant First-Order Logic

The model checking problem for successor-invariant first-order logic on classes of graphs excluding some graph as a minor or as a topological minor is fixed-parameter tractable. This was proved in a series of papers by Engemann, Kreutzer and Siebertz (LICS 2012, [EKS12]), E., Kawarabayashi and Kreutzer (LICS 2013, [EKK13]) and E. and Kawarabayashi (CSL 2016, [EK16]). We proceed in two steps:

1. Model checking for successor-invariant first-order logic can be reduced to model checking for plain FO by adding an arbitrary successor relation to the input structure. The problem is, however, that this additional successor relation might destroy structural properties (such as planarity) of the input structure that are exploited by efficient model checking algorithms. In Section 7.1 We show that it is sufficient to add not a successor-relation (i.e. a Hamiltonian path) but a  $k$ -walk, i.e. a walk that visits every vertex at least once and at most  $k$  times, for some fixed  $k$ .
2. It remains to show how  $k$ -walks can be added to graphs taken from a tame graph class  $\mathcal{C}$  without losing tameness. We do this in Section 7.2 for graphs excluding some graph as a minor, and in Section 7.3 for graphs excluding some graph as a topological subgraph.

Taking a slightly different point of view, the existence of a  $k$ -walk is equivalent to the existence of a spanning tree of maximum degree  $k$ . Using this approach, van den Heuvel et al. [vdHKP<sup>+</sup>17] extended our model checking results to classes of graphs of bounded expansion.

Putting these steps together we obtain the following theorem, which will be proved in Section 7.4.

**Theorem 7.0.4.** *There is an algorithm  $\mathbb{A}$  which takes as input*

- *a finite graph  $H$ ,*
- *a finite  $\sigma$ -structure  $A$  over some relational vocabulary  $\sigma$ , such that the Gaifman graph of  $A$  does not contain  $H$  as a topological subgraph (or, a fortiori, as a minor), and*
- *a successor-invariant formula  $\varphi \in \text{succ-inv-FO}$*

*and checks whether*

$$A \models \varphi$$

*in time  $f(|V(H)| + |\varphi|) \cdot |V(A)|^c$  for some computable function  $f$  and  $c \in \mathbb{N}$ .*

## 7.1 Interpreting a Successor Relation

Suppose we are given a planar graph  $G = (V, E)$  and a sentence  $\varphi \in \text{succ-inv-FO}$  and want to decide whether  $G \models \varphi$ . If  $\varphi$  was a plain FO sentence (without the invariantly used successor relation) we could use known model checking algorithms such as those by Frick and Grohe [FG01] or Dvořák et al. [DKT10]. Since  $\varphi$  uses the successor relation invariantly, it suffices to find *some* linear order  $v_1, \dots, v_n$  of  $V$  such that adding the edges

$$v_1v_2, v_2v_3, \dots, v_{n-1}v_n$$

to  $G$  preserves planarity. However, this is known to be impossible in general (cf. [MM63]).

We simplify the problem by showing that it suffices to add not a Hamiltonian path (i.e. a path visiting each vertex exactly once), but a  $k$ -walk, i.e. a walk visiting each vertex at least once and at most  $k$  times.

**Definition 7.1.1** ( $k$ -walk). For  $k \geq 1$ , a  $k$ -walk  $w$  of length  $\ell$  through a graph  $G = (V, E)$  is a function  $w : [\ell] \rightarrow V$  such that

- $w$  is surjective,
- $w(i)w(i+1) \in E$  for  $i = 1, \dots, \ell - 1$ , and
- $|w^{-1}(v)| \leq k$  for all  $v \in V$ .

We show that a successor relation is FO-interpretable from a  $k$ -walk in the following sense:

**Lemma 7.1.2.** *Let  $\sigma$  be a finite relational signature,  $A$  a finite  $\sigma$ -structure, and  $w : [n] \rightarrow V(A)$  a  $k$ -walk through the Gaifman graph of  $A$ , where  $n \leq k \cdot |V(A)|$ .*

*Then there is a finite relational signature  $\sigma_k$  and a first-order formula  $\varphi_{\text{succ}}^{(k)}(x, y)$ , both depending only on  $k$ , and a  $(\sigma \cup \sigma_k)$ -expansion  $A'$  of  $A$  which can be computed from  $A$  and  $w$  in polynomial time, such that*

- *the Gaifman-graphs of  $A'$  and  $A$  are the same;*
- *$\varphi_{\text{succ}}^{(k)}$  defines a successor relation on  $A'$ .*

*Proof.* We define a function  $f : [n] \rightarrow [k]$  which counts how many times we have visited a vertex on the walk before, by

$$f(i) := |\{j \leq i \mid w(i) = w(j)\}|.$$

Furthermore, let  $F : V(A) \rightarrow [k]$  count how many times we visit a vertex:

$$F(v) := |\{i \in [n] \mid w(i) = v\}|.$$

To simplify notation, if  $i \in [n]$  we write  $F(i)$  for  $F(w(i))$ .

We encode the  $k$ -walk  $w$  by binary relations  $E_{ab}$  with  $a, b = 1, \dots, k$ , in such a way that  $(u, v) \in E_{ab}$  if and only if there is some  $i \in [n-1]$  such that

- $w(i) = u$  and  $f(i) = a$ , and
- $w(i+1) = v$  and  $f(i+1) = b$ .

That is, after visiting  $u$  for the  $a$ -th time, the walk  $w$  proceeds to  $v$ , visiting it for the  $b$ -th time. Note that if  $k = 1$ , we can immediately define a successor relation by

$$\varphi_{\text{succ}}^{(1)}(x, y) := E_{11}xy.$$

If  $k > 1$ , we show how to interpret a  $(k-1)$ -walk  $w'$  in first-order logic, given a  $k$ -walk encoded by  $\{E_{ab} \mid 1 \leq a, b \leq k\}$  as above. By daisy-chaining these interpretations we end up with a 1-walk (i.e. a Hamiltonian path). Plugging in the interpretation of this Hamiltonian path into  $\varphi_{\text{succ}}^{(1)}$  defined above gives the formulae  $\varphi_{\text{succ}}^{(k)}$ .

In order to get from a  $k$ -walk to a  $(k-1)$ -walk, we look at all vertices that are visited  $k$  times, and “jump” over these vertices, either when they are visited for the  $(k-1)$ -st or for the  $k$ -th time. Jumping over a vertex can be done in first-order logic, but we must be careful to choose the vertices for jumping in such a way that we never jump over an unbounded number of vertices in a row, as this is not possible in first-order logic. We encode

the information on whether to jump when visiting for the  $(k-1)$ -th or the  $k$ -th time in a new unary predicate  $P_k$ .

To be precise, let  $\varphi_{k\text{-times}}(x)$  be a formula which states that  $x$  is visited  $k$  times:

$$\varphi_{k\text{-times}}(x) := \bigvee_{a=1}^k \exists y E_{ka}xy.$$

For those  $u \in V(A)$  which are visited  $k$ -times, we agree to jump over them when they are visited for the  $k$ -th time if  $u \in P_k$ , and when they are visited for the  $(k-1)$ -th time otherwise. Thus, if  $w(i) = u$ ,  $f(i) = k$  and  $u \in P_k$ , we want to remove the  $i$ -th step. However, it may be the case that  $w(i+1)$  is also visited  $k$  times and needs to be jumped over. We define first-order formulae which carry out a bounded number of such jumps as follows.

- For  $a \in [k]$ , the formula  $\varphi_{\text{jump},a}(x)$  holds if we jump over  $x$  when visiting it for the  $a$ -th time:

$$\begin{aligned} \varphi_{\text{jump},1}(x), \dots, \varphi_{\text{jump},k-2}(x) &:= \perp, \\ \varphi_{\text{jump},k-1}(x) &:= \varphi_{k\text{-times}}(x) \wedge \neg P_kx, \\ \varphi_{\text{jump},k}(x) &:= \varphi_{k\text{-times}}(x) \wedge P_kx. \end{aligned}$$

- For  $r \geq 0$  and  $a, b \in [k]$ , the formula  $\varphi_{\text{next},a,b}^{(r)}(x, y)$  holds if, when applying at most  $r$  consecutive jumps on entering  $x$  for the  $a$ -th time, we end up in node  $y$  which is visited for the  $b$ -th time in the (original) walk. Specifically:

$$\begin{aligned} \varphi_{\text{next},a,b}^{(0)}(x, y) &:= x \dot{=} y \wedge \delta_{ab}, \\ \varphi_{\text{next},a,b}^{(r+1)}(x, y) &:= (\neg \varphi_{\text{jump},a}(x) \rightarrow (x \dot{=} y \wedge \delta_{ab})) \\ &\quad \wedge \left( \varphi_{\text{jump},a}(x) \rightarrow \exists z \bigvee_{c=1}^k (E_{ac}xz \wedge \varphi_{\text{next},c,b}^{(r)}(z, y)) \right). \end{aligned}$$

Here,  $\delta_{ab}$  is true if the indices  $a$  and  $b$  are the same:

$$\delta_{ab} := \begin{cases} \top, & \text{if } a = b; \\ \perp, & \text{otherwise.} \end{cases}$$

- We will show below how to choose the predicate  $P_k$  so that we never need to take more than two consecutive jumps. Thus, we can interpret

a  $(k-1)$ -walk  $w'$  using, for  $a, b \in [k-2]$ , the formulae

$$\varphi_{E,a,b}(x, y) := \exists z \bigvee_{c=1}^k (E_{ac}xz \wedge \varphi_{\text{next},c,b}^{(2)}(z, y)).$$

For  $a \in [k-2]$  we set

$$\varphi_{E,a,k-1}(x, y) := \exists z \bigvee_{c=1}^k (E_{ac}xz \wedge (\varphi_{\text{next},c,k-1}^{(2)}(z, y) \vee \varphi_{\text{next},c,k}^{(2)}(z, y))).$$

Next, for  $b \in [k-2]$  we set

$$\begin{aligned} \varphi_{E,k-1,b}(x, y) := & \left( \neg \varphi_{\text{jump},k-1}(x) \rightarrow \exists z \bigvee_{c=1}^k (E_{k-1,c}xz \wedge \varphi_{\text{next},c,b}^{(2)}(z, y)) \right) \\ & \wedge \left( \varphi_{\text{jump},k-1}(x) \rightarrow \exists z \bigvee_{c=1}^k (E_{k,c}xz \wedge \varphi_{\text{next},c,b}^{(2)}(z, y)) \right), \end{aligned}$$

and finally we define

$$\begin{aligned} \varphi_{E,k-1,k-1}(x, y) := & \left( \neg \varphi_{\text{jump},k-1}(x) \rightarrow \right. \\ & \exists z \bigvee_{c=1}^k (E_{k-1,c}xz \wedge (\varphi_{\text{next},c,k-1}^{(2)}(z, y) \vee \varphi_{\text{next},c,k}^{(2)}(z, y))) \\ & \wedge \left( \varphi_{\text{jump},k-1}(x) \rightarrow \right. \\ & \left. \left. \exists z \bigvee_{c=1}^k (E_{k,c}xz \wedge (\varphi_{\text{next},c,k-1}^{(2)}(z, y) \vee \varphi_{\text{next},c,k}^{(2)}(z, y))) \right) \right). \end{aligned}$$

To define the predicate  $P_k$ , let  $T \subseteq [n]$  be the set of indices  $i \in [n]$  for which  $F(i) = k$  and  $f(i) \in \{k-1, k\}$ . We obtain a perfect matching  $M$  on  $T$  by matching  $i$  and  $j$  if and only if  $w(i) = w(j)$  (cf. Figure 7.1 (a)). We define a subset  $J \subset [n]$  with the intended meaning that if  $i \in J$ , we jump over the  $i$ -th step of  $w$ . The set  $J$  will satisfy the following two conditions:

- every vertex  $v$  with  $F(v) = k$  is jumped over exactly once, i.e.

$$|\{i \in [n] \mid w(i) = v\} \cap J| = 1, \quad \text{and}$$

- we never jump more than twice in a row, i.e. if  $i, i+1 \in J$ , then  $i+2 \notin J$ .



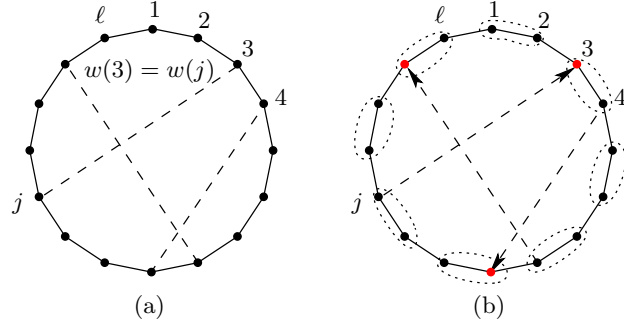


Figure 7.1: Deciding when to jump over vertices in a  $k$ -walk.

We partition the set  $[n]$  into intervals of size 2, setting

$$U := \{\{1, 2\}, \{3, 4\}, \dots\},$$

with the last set  $\{n\}$  being a singleton if  $n$  is odd. Then the matching  $M$  defines a multigraph without loops on  $U$ , and the degree of each  $I \in U$  is at most 2. We direct the edges of  $M$ , viewed as edges in the multigraph  $(U, M)$ , in such a way that every  $I \in U$  has at most one incoming edge. The edges incident with  $I$  correspond to the elements of  $I \cap T$ , and we put  $i \in I$  into  $J$  if and only if the edge corresponding to  $i$  is directed towards  $I$  (cf. Figure 7.1 (b)). For every  $k = 1, \dots, \lfloor \frac{1}{2}(n-1) \rfloor$  at most one of  $2k-1$  and  $2k$  is in  $J$ , and therefore  $J$  satisfies the above requirements.

The definition of  $P_k \subseteq V(G)$  is now straightforward:

$$P_k := \{v \in V(G) \mid F(v) = k \text{ and } f(i) = k \text{ for the } i \in J \text{ with } w(i) = v\}.$$

In summary, we end up with

$$\sigma_k := \{E_{ab} \mid a, b \in [k]\} \cup \{P_a \mid a = 2, \dots, k\},$$

and it is clear that our construction can be carried out in polynomial time.  $\square$

## 7.2 $k$ -Walks in Graphs with Excluded Minors

In this section we show that  $k$ -walks can be added to graphs excluding some minor while preserving the property that some (possibly larger) clique is still excluded as a minor. Our construction is based on topological considera-

tions, Robertson and Seymour's structure theorem for graphs with excluded minors [RS03], and its algorithmic version by Demaine et al. [DHK05].

**Lemma 7.2.1.** *For every natural number  $r$  there is a  $k = k(r)$  such that: If  $G = (V, E)$  is a graph which does not contain a  $K_r$ -minor, then there is a supergraph  $G' = (V, E')$  obtained from  $G$  by possibly adding edges such that  $G'$  does not contain a  $K_k$ -minor and there is a  $k$ -walk  $w$  through  $G'$ . Moreover,  $G'$  and  $w$  can be found in polynomial time for fixed  $r$ .*

*Proof.* Without loss of generality we assume that  $G$  is connected. We first compute a tree-decomposition  $(\mathcal{T}, \mathcal{V})$  of  $G$  whose torsos are  $s$ -nearly embeddable into some surface into which  $K_r$  can not be embedded, for some  $s$  depending only on  $r$ . Such a decomposition exists by the Graph Structure Theorem (Thm. 3.4.2), and it can be computed in polynomial time for fixed  $r$  (cf. [GKR13, DHK05]).

Assume that each bag  $\mathcal{V}$  comes with an  $s$ -near embedding of its torso, i.e., for each bag  $\mathcal{V}_t$  we are given a set  $Z_t$  of at most  $s$  apices, subgraphs  $\mathcal{V}_t^{(0)}, \dots, \mathcal{V}_t^{(s)}$  of  $\bar{\mathcal{V}}_t \setminus Z_t$ , and an embedding  $\Pi_t$  of the graph  $\mathcal{V}_t^{(0)}$  into a surface into which  $K_r$  can not be embedded and such that  $\mathcal{V}_t^{(1)}, \dots, \mathcal{V}_t^{(s)}$  are attached as vortices to this embedding. The algorithm in [DHK05] actually yields a decomposition and embeddings for which the tree  $\mathcal{T}$  is rooted, say, with root  $t_r$ , and such that for every pair of nodes  $t$  and  $u$  such that  $u$  is a child of  $t$  we have

$$\mathcal{V}_t \cap \mathcal{V}_u \subseteq Z_u \tag{7.1}$$

and  $(\mathcal{V}_t \cap \mathcal{V}_u) \setminus Z_t$  is either contained in a single bag of the path-decomposition of one  $\mathcal{V}_t^{(i)}$  for  $i \geq 1$  or is a set of size at most three and the vertices in this set lie on the boundary of a face of  $\Pi_t$ . By adding edges to  $G$  if necessary we may assume that

- (D1) all bags are identical to their torsos, i.e.,  $\mathcal{V}_t = \bar{\mathcal{V}}_t$  for all  $t \in \mathcal{T}$ ,
- (D2) in every bag  $t \in T$ , all apices  $z \in Z_t$  are connected to all other vertices in  $\mathcal{V}_t$ ,
- (D3) for every  $t, u \in T$  such that  $u$  is a child of  $t$ , the set  $(\mathcal{V}_t \cap \mathcal{V}_u) \setminus Z_t$  is either a clique contained in a single bag of the path-decomposition of one  $\mathcal{V}_t^{(i)}$  for  $i \geq 1$  or a face of  $\Pi_t$  of size three (i.e., a triangle).

We will need these properties later. After adding these edges the graph still excludes some clique as a minor; for a proof cf. [JW13, Thm. 1.1]. We will keep adding edges to  $G$  in the course of this proof. For ease of notation, we still call the resulting supergraphs  $G$ .

We add chords to all facial cycles of the embedding  $\Pi_t$  to turn it into a triangulation. In particular, the resulting graph is 3-connected (note that the neighbourhood of every  $v \in \mathcal{V}_t^{(0)}$  induces a cycle as a subgraph, which is 2-connected), and since it is still embedded into the same surface as before we did not create a  $K_r$ -minor. By induction on the Euler genus  $\text{eg}$  of  $\Pi_t$  (cf. [MT01]) we show that there is a  $2^{\text{eg}+1}$ -walk through  $\mathcal{V}_t^{(0)}$  which can be computed in polynomial time.

The base case for the induction is  $\text{eg} = 0$ , i.e., if  $\mathcal{V}_t^{(0)}$  is a planar graph. In this case we invoke a result by Gao and Richter [GR94], which states that every 3-connected planar graph contains a 2-walk. Note that for every  $k \geq 1$  the 2-connected planar graph  $K_{2,2k}$  does not have a  $k$ -walk, so 3-connectivity is essential here, and there are 3-connected planar graphs without Hamiltonian paths (cf. [MM63]). On the other hand, 4-connected planar graphs are known to have Hamiltonian paths (i.e. 1-walks) by a result of Tutte [Tut56, Tho83].

While Gao and Richter do not mention it explicitly, by inspection of the proof in [GR94] we immediately find that such a 2-walk can be computed in polynomial time. Note that we extended  $\mathcal{V}_t^{(0)}$  to a triangulation (which is 3-connected) exactly to be able to apply this theorem.

For the induction step, assume that  $\mathcal{V}_t^{(0)}$  is not planar. In this case it contains non-contractible cycles, and we compute a shortest such cycle  $C$ ; this can be done in polynomial time, cf. [Tho90]. We need to distinguish two cases.

First assume  $C$  is two-sided and surface separating. Define left and right edges adjacent to  $C$ , as well as the left and right subgraph  $G_l$  and  $G_r$ , as in [MT01, Ch. 3]. Since  $C$  is surface separating, the graphs  $G_l$  and  $G_r$  are distinct, and by [MT01, Prop. 4.2.1] the Euler genus of  $\Pi_t$  equals the sum of the Euler genera of the induced embeddings  $\Pi_l$  and  $\Pi_r$  of  $G_l \cup C$  and  $G_r \cup C$ . Since we assumed  $C$  to be non-contractible, neither of these graphs is planar, and so both have strictly positive Euler genus, which is strictly smaller than the Euler genus of  $\Pi_t$ . By induction, we find  $2^{\text{eg}}$ -walks  $w_l$  and  $w_r$  in  $G_l \cup C$  and  $G_r \cup C$ . Putting these together and joining them at some vertex in  $C$  yields a  $2 \cdot 2^{\text{eg}} = 2^{\text{eg}+1}$ -walk in  $\mathcal{V}_t^{(0)}$ .

Otherwise, if  $C$  is two-sided but not surface separating, or if  $C$  is one-sided, we cut along  $C$  as in [MT01, Lemma 4.2.4]. This results in a graph  $H$  whose Euler genus is lower than that of  $\mathcal{V}_t^{(0)}$  and such that each vertex of  $C$  has two copies in  $H$ . Again, we use induction to find a  $2^{\text{eg}}$ -walk in  $H$ , which directly gives us a  $2 \cdot 2^{\text{eg}} = 2^{\text{eg}+1}$ -walk in  $\mathcal{V}_t^{(0)}$ .

In order to extend this walk to a walk through  $\mathcal{V}_t^{(0)} \cup \mathcal{V}_t^{(1)} \cup \dots \cup \mathcal{V}_t^{(s)}$ , we add edges to each  $\mathcal{V}_t^{(i)}$ ,  $i = 1, \dots, s$ , to turn each bag of the path-decomposition of  $\mathcal{V}_t^{(i)}$  into a clique. Each bag contains some vertex  $v \in \mathcal{V}_t^{(0)}$ , and the first time the walk through  $\mathcal{V}_t^{(0)}$  enters this  $v$  we make a detour through all nodes in its bag which have not been visited before, return to  $v$ , and then continue the walk. This results in a  $(2^{\text{eg}+1} + 1)$ -walk through  $\mathcal{V}_t^{(0)} \cup \mathcal{V}_t^{(1)} \cup \dots \cup \mathcal{V}_t^{(s)}$ .

So far we have found  $k'$ -walks through the torsos (excluding the apices) of our tree-decomposition, for some  $k'$  depending only on  $r$ . We added some edges, but the torsos still exclude some clique  $K_{r'}$  as a minor: When we added chords to turn  $\mathcal{V}_t^{(0)}$  into a triangulation, we also added chords through the cuffs at which the  $\mathcal{V}_t^{(i)}$  are glued, but this can be done by connecting *one* node  $v_i$  on the cuff  $C_i$  to all other nodes on  $C_i$ , and by adding  $v$  to all bags of the path-decomposition of  $\mathcal{V}_t^{(i)}$  we still get an almost-embeddable graph, which excludes some clique minor.

Furthermore, we only added edges within bags of our tree-decomposition  $(\mathcal{T}, \mathcal{V})$ , and these edges do not affect other bags, because we assumed all bags to be identical to their torsos: If  $u, v \in V$  appear together in more than one bag, then there is already an edge between them in  $G$ . Thus  $(\mathcal{T}, \mathcal{V})$  remains a tree-decomposition even after adding edges.

Now we extend these walks through the apices  $Z_t$  of each bag and paste the walks in the individual bags together to get a walk through the whole graph. We start at the root  $t_r$  of  $(\mathcal{T}, \mathcal{V})$  and pick, for every  $z \in Z_{t_r}$ , an arbitrary neighbour  $v \in \mathcal{V}_{t_r}^{(0)}$  of  $z$ . If there is no such neighbour we add an edge to an arbitrary node. We make a detour through  $z$  the first time the walk visits  $v$ . This increases the number of times we visit  $v$  by one, but since  $|Z_t| < s$ , it can only turn our  $k'$ -walk into a  $k_1 := (k' + s)$ -walk.

We extend the  $k_1$ -walk step by step until it covers the whole graph  $G$ . Let  $\tilde{\mathcal{T}}$  be the set of nodes of the tree-decomposition which are already covered, i.e., such that we already have a walk through  $\bigcup_{t \in \tilde{\mathcal{T}}} \mathcal{V}_t$ . By the last paragraph we may start with  $\tilde{\mathcal{T}} = \{t_r\}$ . Let  $\partial\tilde{\mathcal{T}}$  be the set of nodes  $t \in \tilde{\mathcal{T}}$  which still have children  $u \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ . In each step we pick one  $t \in \partial\tilde{\mathcal{T}}$  and extend the walk through all its children. The constructed walk will be a  $k_2$ -walk for some  $k_2 > k_1$ , but for every  $t \in \partial\tilde{\mathcal{T}}$ , no vertex from  $\mathcal{V}_t \setminus Z_t$  will be visited more than  $k_1$  times.

Let  $t \in \partial\tilde{\mathcal{T}}$  be a bag with children  $u_1, \dots, u_n \in \mathcal{T}$ . We call

$$A_i := \mathcal{V}_{u_i} \cap \mathcal{V}_t$$

the *adhesion set* of  $u_1$ . The basic idea is to insert the  $k_1$ -walk  $w_i$  through the bag  $u_i$  into the walk  $w$  through  $t$  when the latter visits some vertex  $v_i \in A_i$ . However, inserting  $w_i$  at  $v_i$  increases the number of times we visit  $v_i$  by one, and since the number of children  $n$  is unbounded, we have to carefully choose the vertices  $v_i \in A_i$  so that no single vertex is used more than a bounded number of times.

We first show that we may assume the adhesion sets  $A_i$  to be distinct. By Property (D7.2), each  $A_i$  is of the form

$$A_i = A_i^{(Z)} \cup \Delta_i,$$

where  $A_i^{(Z)} := A_i \cap Z_t$  and  $\Delta_i$  is a face of  $\Pi_t$  (the embedding of  $\mathcal{V}_t^{(0)}$ ) or a clique which is contained in some bag of a path-decomposition of a vortex. It may well happen that an unbounded number of children of  $t$  have the same adhesion set, and we first show how to deal with this case.

Let  $u_1$  be a child of  $t$  with adhesion set  $A$ , and let  $u_2, \dots, u_m$  be the other children of  $t$  with the same adhesion set. For each  $u_i$  we pick the endpoints  $a_i, b_i \in \mathcal{V}_{u_1}^{(0)}$  of an arbitrary edge traversed from  $a_i$  to  $b_i$  by the walk  $w_i$ . (The case in which  $\tilde{V}_{u_i}^{(0)}$  does not contain an edge can be dealt with easily.) We add edges  $a_1b_2, a_2b_3, \dots, a_mb_1$  to  $G$  as in Figure 7.2. The resulting graph still excludes some clique, for if  $r' \geq |A| + 3$  and none of the bags  $u_i$  has a  $K_{r'}$  minor, then neither does  $\bigcup_{i=1}^m \mathcal{V}_{u_i}$  with the added edges. This follows from Lemma 3.4.3, because  $\mathcal{V}_{u_i}$  intersects  $\bigcup_{j \neq i} \mathcal{V}_{u_j}$  in  $A \cup \{a_i, b_i\}$ , which is a clique of size  $|A| + 2$ .

We replace the bags  $u_1, \dots, u_m$  by a single new bag  $\tilde{u}$ . While  $\tilde{u}$  is no longer nearly embedded, it still excludes some clique minor (whose size depends only on  $r$ ).

Using the new edges we connect the walks  $w_1, \dots, w_m$  through the bags  $u_i, \dots, u_m$  as follows: By our choice of  $a_i$  and  $b_i$ , each walk  $w_i$  traverses the edge  $a_ib_i$  at some point. We go from  $a_i$  to  $b_{i+1}$  instead (and from  $a_m$  to  $b_1$ ). The resulting walk  $\tilde{w}$  is a  $k_1$ -walk through  $\tilde{u}$ .

From now on we assume that no two children of  $t$  have identical adhesion sets. We want to pick a  $v_i \in A_i$  such that no  $v \in \mathcal{V}_t$  is chosen more than a bounded number of times. Having done so we can insert the walk  $w_i$  into the walk through  $t$  at  $v_i$  without increasing the number of times  $v_i$  is visited by too much.

We have to distinguish two cases: If  $\Delta_i$  is contained in one bag of a path-decomposition of one of the vortices of  $t$ , we use the indexing vertex

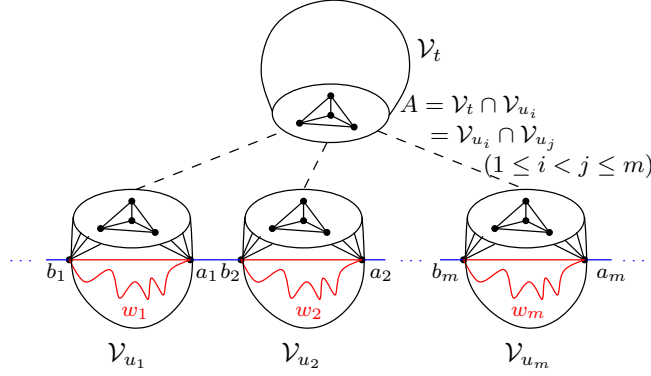


Figure 7.2: The walks  $w_1, \dots, w_m$  are joined by replacing edges  $a_i b_i$  on the walks (red) by edges  $a_i b_{i+1}$  (blue).

of that bag as  $v_i$ . By our bound on the path-width of the vertices, no  $v_i$  is used more than a bounded number of times.

For  $\Delta_i$  which form a face of the embedding of  $\mathcal{V}_t^{(0)}$  we proceed as follows: Let  $\hat{G}$  be a barycentric subdivision of  $(\mathcal{V}_t^{(0)}, \Pi_t)$ , i.e., we introduce a new vertex  $v_F$  for every face  $F$  of  $\Pi_t$  and connect it to  $v \in V(\mathcal{V}_t^{(0)})$  iff  $v \in F$ . Then  $\hat{G}$  is again 3-connected and  $\Pi_t$  can be extended to an embedding of  $\hat{G}$  in an obvious way. We compute a  $2^{\text{eg}+1}$ -walk  $\hat{w}$  through  $\hat{G}$  as above. Each  $v_F$  is visited at least once, and since all its neighbours are vertices of  $\mathcal{V}_t^{(0)}$ , its immediate predecessor on  $\hat{w}$  when it is first visited is some  $u_F \in \mathcal{V}_t^{(0)}$ . Now if  $\Delta_i$  is some face  $F$  of  $\Pi_t$ , we insert  $w_i$  into  $w$  when first visiting  $u_F$ . This way  $u_F$  is used at most  $2^{\text{eg}+1}$  times.  $\square$

### 7.3 $k$ -Walks in Graphs with Excluded Topological Subgraphs

Using our construction of  $k$ -walks in graphs with excluded minors and the structure theorem for graphs with excluded topological subgraphs (Theorem 3.4.5), we show that  $k$ -walks can be added to such graphs in a way that still guarantees some graph to be excluded as a topological subgraph. These results have been presented in [EK16].

**Lemma 7.3.1.** *For every finite graph  $H$  there are constants  $k \in \mathbb{N}$  and  $c \in \mathbb{N}$  such that for every graph  $G$  which does not contain  $H$  as a topological subgraph there is a graph  $G'$  and a  $k$ -walk  $w : [\ell] \rightarrow V(G')$  through  $G'$  such that  $G'$  is obtained from  $G$  by only adding edges and  $G'$  does not contain  $K_c$*

as a topological subgraph. Furthermore,  $k$ ,  $c$ ,  $G'$  and  $w$  can be computed, given  $G$  and  $H$ , in time  $f(|V(H)|) \cdot |V(G)|^d$  for some computable function  $f$  and  $d \in \mathbb{N}$ .

For the rest of this section we assume a graph  $G = (V, E)$  together with a tree-decomposition  $(\mathcal{T}, \mathcal{V})$  satisfying the properties of Theorem 3.4.5 as given. We will construct  $k$ -walks through each of the bags of this decomposition, for a suitable  $k$  depending only on  $H$ , suitably adding edges within the bags in a way that will not create large topological subgraphs. We will then connect these  $k$ -walks to obtain a  $k'$ -walk through all of  $G$ , carefully adding further edges where necessary.

If  $s, t \in \mathcal{T}$  are neighbours in  $\mathcal{T}$  we will connect the  $k$ -walk through  $\mathcal{V}_s$  and the  $k$ -walk through  $\mathcal{V}_t$  by joining them along a suitably chosen vertex  $v \in \mathcal{V}_s \cap \mathcal{V}_t$ . Since the resulting walk may visit  $v$  a total of  $k + 1$  times, we must be careful not to select the same vertex  $v$  more than a bounded number of times.

We first pick an arbitrary tree node  $r \in \mathcal{T}$  as the root of the tree-decomposition. Notions such as parent and sibling nodes are meant with respect to this root node  $r$ . For a node  $t \in \mathcal{T}$  we define its adhesion set  $\alpha_t \subseteq \mathcal{V}_t$  as

$$\alpha_t := \begin{cases} \emptyset & \text{if } t = r \\ \mathcal{V}_s \cap \mathcal{V}_t & \text{if } s \text{ is the parent of } t. \end{cases}$$

By adding the necessary edges within the bags we may assume that each  $\mathcal{V}_t$  is identical to its torso, in other words we may assume that  $G[\alpha_t]$  is a clique for each  $t \in \mathcal{T}$ .

### Computing the $k$ -walks $w_t$

Let  $s, t \in \mathcal{T}$  be nodes such that  $s$  is the parent of  $t$ . It may happen that  $\alpha_s \cap \alpha_t \neq \emptyset$ , and in fact we can not bound

$$|\{s \in \mathcal{T} \mid v \in \mathcal{V}_s\}|$$

for all  $G$  excluding a fixed topological subgraph and all  $v \in V(G)$ . Since we are only allowed to visit each vertex a bounded (for a fixed excluded topological subgraph) number of times, we first compute, for  $t \in \mathcal{T}$ , a  $k$ -walk  $w_t$  through a suitable supergraph of  $\mathcal{V}_t \setminus \alpha_t$ .

If  $\bar{\mathcal{V}}_t$  contains only  $c$  vertices of degree larger than  $c$  we choose an arbitrary enumeration  $v_1, \dots, v_\ell$  of  $\mathcal{V}_t \setminus \alpha_t$  and add edges

$$v_1v_2, v_2v_3, \dots, v_{\ell-1}v_\ell, v_\ell v_1$$

to  $G$  as far as they are not already present. This will increase the degree of each vertex by at most 2, so there are still at most  $c$  vertices of degree larger than  $c + 2$ . We set

$$\begin{aligned} w_t : [\ell] &\rightarrow \mathcal{V}_t \\ i &\mapsto v_i \end{aligned}$$

for these bags.

If, on the other hand,  $\bar{\mathcal{V}}_t$  excludes a clique  $K_c$  as a minor, we invoke Lemma 7.2.1 on the graph  $\mathcal{V}_t \setminus \alpha_t$ , i.e. we compute a  $k$ -walk  $w_t$  through a supergraph of  $\mathcal{V}_t \setminus \alpha_t$  obtained by adding edges in a way that this supergraph still does not contain a  $K_{c'}$ -minor. Since we ignore the vertices in  $\alpha_t$  when computing the  $k$ -walk  $w_t$ , it may happen that the resulting supergraph of  $\bar{\mathcal{V}}_t$  *does* contain a  $K_{c'}$ -minor. However, the largest possible clique minor is still of bounded size, because  $|\alpha_t| \leq c$ :

**Lemma 7.3.2.** *Let  $G = (V, E)$  be a graph such that  $K_{c'} \not\leq G$ , and let  $G \oplus K_c$  be the graph with vertex set  $V' = V \cup [c]$  and edge set*

$$E' = E \cup \binom{[c]}{2} \cup \{va \mid v \in V, a \in [c]\}.$$

*In other words,  $G \oplus K_c$  is the disjoint sum of  $G$  and  $K_c$  plus edges between all vertices of  $G$  and all vertices of  $K_c$ . Then  $K_{c+c'} \not\leq G \oplus K_c$ .*

*Proof.* Otherwise let  $X_1, \dots, X_{c+c'}$  be the branch sets of a  $K_{c+c'}$ -minor in  $G \oplus K_c$ . At most  $c$  of the sets contain vertices of the added  $K_c$ -clique. The remaining sets form the branch sets of a  $K_{c'}$ -minor in  $G$ , contradicting the assumption that  $K_{c'} \not\leq G$ .  $\square$

## Connecting the $k$ -walks

We still need to connect the  $k$ -walks through the individual bags of  $(\mathcal{T}, \mathcal{V})$  to obtain a single  $k'$ -walk through the whole graph, for some  $k'$  to be determined below. This is the most complicated part of our construction, since we must guarantee that no vertex is visited more than  $k'$  times by the



resulting walk, and that no large topological clique subgraphs are created.

In the case of graphs excluding some fixed minor, the Graph Structure Theorem guarantees the existence of a tree-decomposition into nearly embeddable graphs such that neighbouring bags intersect only in apices and vertices lying on some face or vortex of their near embeddings, and this was used in Section 7.2 to select vertices from the adhesion sets of bags in a suitable way. Since the decomposition theorem for graphs excluding a topological minor does not provide this kind of information, we need a different approach here. Instead, our method for selecting vertices along which to connect the  $k$ -walks relies on the fact that graphs embeddable on a surface are *degenerate*, i.e. every subgraph of such a graph contains some vertex of small degree.

In connecting the walks  $w_t$ , we will proceed down the tree  $\mathcal{T}$ . At any point in the process we keep a set  $D \subseteq \mathcal{T}$  and a walk  $w$  such that

- $D$  is a connected subset of  $\mathcal{T}$ ,
- the  $k'$ -walk has been constructed in  $\bigcup_{t \in D} \mathcal{V}_t$ ,
- if  $s \in D$  and  $s'$  is a sibling of  $s$  then also  $s' \in D$ ,
- $w$  is a  $k'$ -walk through  $\bigcup_{t \in D} \mathcal{V}_t$ , and if  $s \in D$  has a child  $t \notin D$ , then the vertices in  $\mathcal{V}_s \setminus \alpha_s$  are visited at most  $k + 1$  times by  $w$ .

We start with  $D = \{r\}$  and  $w = w_r$ , where  $r$  is the root of  $\mathcal{T}$ . This is easily seen to satisfy all of the above conditions.

Now let  $s \in D$  be a node whose children  $t_1, \dots, t_n$  are not in  $D$ . We let

$$C_i := \alpha_{t_i} \setminus \alpha_s$$

be the adhesion set of  $t_i$  with all vertices of the adhesion set of  $s$  removed. If  $C_i = \emptyset$  then  $t_i$  can be made a sibling of  $s$  (rather than a child), so we assume that all  $C_i$  are nonempty. Since the properties of  $(\mathcal{T}, \mathcal{V})$  are guaranteed for the torsos of the bags we may assume that  $G[C_i]$  is a clique for each  $i$  and that  $w$  visits the vertices of  $\bigcup C_i$  at most  $k + 1$  times.

It may happen that  $C_i = C_j$  for some  $i \neq j$ . To deal with this, assume that

$$C_1 = C_2 = \dots = C_m \neq C_i \text{ for } i > m.$$

For each  $i = 1, \dots, m$  we choose an edge  $u_i v_i \in E(\mathcal{V}_{t_i})$  which is traversed by the walk  $w_{t_i}$  in the direction from  $u_i$  to  $v_i$  at some point. We add edges

$$u_i v_{i+1} \text{ for } i = 1, \dots, m-1 \text{ and } u_m v_1$$

and connect the walks  $w_{t_1}, \dots, w_{t_m}$  along these edges. Because  $w_{t_i}$  is a walk through  $\mathcal{V}_{t_i} \setminus \alpha_{t_i}$ , we have

$$u_i, v_i \in \mathcal{V}_{t_j} \iff i = j$$

for  $i, j = 1, \dots, m$ . To accommodate for the extra edges, we add the vertices  $u_i$  and  $v_i$  to  $\mathcal{V}_s$ , and therefore to  $\alpha_{t_i}$  and  $C_i$ . These vertices together with the added edges form a cycle

$$u_1 v_1 u_2 v_2 \dots u_m v_m u_1$$

in  $\mathcal{V}_s$  and there are no edges between them and other vertices of  $\mathcal{V}_s$ . Therefore no new topological subgraphs are created by this. The maximal adhesion of  $(\mathcal{T}, \mathcal{V})$  is still bounded by  $c + 2$ .

Therefore we now assume that the cliques  $C_1, \dots, C_n$  are all distinct. It remains to find a function

$$f : [n] \rightarrow V$$

such that

- $f(i) \in C_i$  for all  $i$ , and
- $|f^{-1}(v)| \leq M$  for all  $v \in V$  and some constant  $M$  depending only on  $H$ .

We define the function  $f$  iteratively on larger subsets of  $[n]$  as follows: Let  $\tilde{G}$  be the subgraph of  $G$  induced on the union of all  $C_i$ :

$$\tilde{G} = G \left[ \bigcup_i C_i \right].$$

We show that  $\tilde{G}$  contains a vertex of degree (in  $\tilde{G}$ ) at most  $d$ , for some constant  $d$  depending only on the constant  $c$  from Theorem 3.4.5 (and therefore only on the excluded topological subgraph  $H$  we started with). If  $\mathcal{V}_s$  contains only  $c$  vertices of degree larger than  $c$  then this is true with  $d = c$ . If  $\mathcal{V}_t$  excludes some clique  $K_c$  as a minor we use the fact that these graphs are  $d$ -degenerate for some  $d$  depending only on  $c$ . In fact, by Theorem 7.2.1 in [Die12] there is a constant  $d$  such that if the average degree of  $\tilde{G}$  is at least  $d$ , then  $K_c \preceq_{\text{top}} \tilde{G}$  and therefore  $K_c \preceq \tilde{G}$ .

In both cases there is a  $v \in \bigcup_i C_i$  which has degree at most  $d$  in  $\tilde{G}$ . We want to bound the number of  $i \in [n]$  for which  $v \in C_i$ . Since every clique  $C_i$  has size at most  $c + 2$ , and if  $v \in C_i$  then all elements of  $C \setminus \{v\}$  are

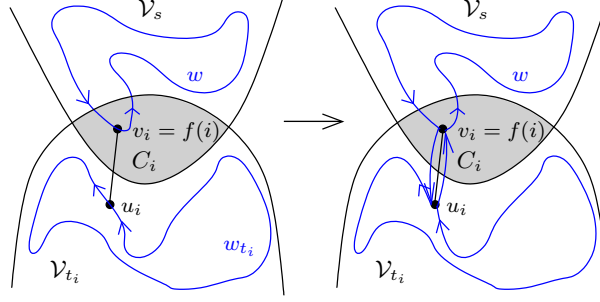


Figure 7.3: Connecting the individual  $k$ -walks

neighbours of  $v$ , there can be at most

$$M := \binom{d}{0} + \binom{d}{1} + \cdots + \binom{d}{c+1}$$

many such  $C_i$ , and this bound only depends on  $c$ . It is therefore safe to define

$$f(i) := v \text{ for all } i \in [n] \text{ such that } v \in C_i.$$

We remove these cliques and iterate until no cliques remain.

Once the function  $f$  has been found we connect the walk  $w$  through  $\bigcup_{t \in D} \mathcal{V}_t$  with the walks  $w_{t_i}$  through the bags  $V_{t_i}$ . Let  $w : [\ell] \rightarrow V$  be the walk constructed so far. For each  $i \in [n]$  let  $v_i = f(i) \in C_i$  be the vertex chosen by  $f$ , and let  $u_i \in \mathcal{V}_{t_i} \setminus \alpha_{t_i}$  be a neighbour of  $v_i$ . If no such neighbour exists it is safe to create one by adding an edge between  $v_i$  and an arbitrary vertex of  $\mathcal{V}_{t_i} \setminus \alpha_{t_i}$ . We now extend the walk  $w$  by inserting the  $k$ -walk  $w_{t_i}$  along the edge  $v_i u_i$  when  $v_i$  is first visited by  $w$ . This increases the number of times  $v_i$  and  $u_i$  are visited by one each (cf. Figure 7.3).

After inserting all walks  $w_{t_1}, \dots, w_{t_n}$  we set

$$D := D \cup \{t_1, \dots, t_n\}$$

and repeat the process until  $D = \mathcal{T}$ . Note that the resulting walk is a  $(k + M + 1)$ -walk through the supergraph  $G'$  of  $G$  obtained by adding edges to  $G$ .

## Topological Subgraphs in $G'$

By now we have a supergraph  $G'$  of  $G$ , obtained by only adding edges, and a  $k' = (k + M + 1)$ -walk  $w : [\ell] \rightarrow V(G')$  through this supergraph. Furthermore, by Thm. 3.4.5 there is a  $c' = c'(H)$  depending only on (the size of)  $H$  and a tree-decomposition  $(\mathcal{T}, \mathcal{V})$  of  $G'$  such that if  $s, t \in \mathcal{T}$  then  $|\mathcal{V}_s \cap \mathcal{V}_t| \leq c'$  and for all  $t \in \mathcal{T}$

- $\bar{\mathcal{V}}_t$  has at most  $c'$  vertices of degree larger than  $c'$  or
- $\bar{\mathcal{V}}_t$  excludes  $K_{c'}$  as a minor.

We show that this implies  $K_{c'+2} \not\preceq_{\text{top}} G'$ : Assume for a contradiction that  $K_{c'+2} \preceq_{\text{top}} G$ , and let  $v_1, \dots, v_{c'+2}$  be the branch vertices of a  $K_{c'+2}$ -subdivision in  $G$ . Then there is a  $t \in \mathcal{T}$  such that  $\{v_1, \dots, v_{c'+2}\} \subseteq \mathcal{V}_t$ : Otherwise choose  $i < j$  and  $t \neq t'$  so that

$$v_i \in \mathcal{V}_t \setminus \mathcal{V}_{t'} \quad \text{and} \quad v_j \in \mathcal{V}_{t'} \setminus \mathcal{V}_t.$$

Then, since the adhesion of  $(\mathcal{T}, \mathcal{V})$  is at most  $c'$ , there is a set  $S \subseteq V$  of size at most  $c'$  separating two branch vertices, which is not possible in a  $(c' + 2)$ -clique.

Now let  $t \in \mathcal{T}$  be a tree node for which  $\mathcal{V}_t$  contains all branch vertices. For  $i < j$ , let  $P_{ij}$  be the path in  $G$  connecting  $v_i$  and  $v_j$ . If all vertices on this path are in  $\mathcal{V}_t$  we are done. Otherwise we may shorten this path to get a path  $P'_{ij}$  connecting  $v_i$  and  $v_j$  in the torso of  $\mathcal{V}_t$ . Thus

$$K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t.$$

But none of the bags  $\mathcal{V}_t$  can contain  $K_{c'+2}$  as a topological subgraph: Since  $K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t$  implies  $K_{c'+2} \preceq \mathcal{V}_t$  which in turn implies  $K_{c'} \preceq \mathcal{V}_t$ , none of the bags excluding  $K_{c'}$  as a minor can contain  $K_{c'+2}$  as a topological subgraph. But if  $K_{c'+2} \preceq_{\text{top}} \mathcal{V}_t$  then there must be at least  $c' + 2$  vertices of degree at least  $c' + 1$ , namely the branch vertices of the image of a subdivision of  $K_{c'+2}$ . We conclude that  $K_{c'+2} \not\preceq_{\text{top}} G'$ .

## 7.4 The Model Checking Algorithm

*Proof of Theorem 7.0.4.* Given a  $\sigma$ -structure  $A$ , a successor-invariant formula  $\varphi \in \text{succ-inv-FO}(\sigma)$  and a graph  $H$  which is not a topological subgraph of the Gaifman graph of  $A$ , we first compute the Gaifman graph  $G$  of  $A$ . Using the algorithm of Lemma 7.3.1 we then compute a  $k$ -walk  $w : [\ell] \rightarrow V(A)$

through a supergraph  $G'$  of  $G$  which excludes some clique  $K_{c'}$  as a topological subgraph.

Let  $E$  be a binary relation symbol. We expand  $A$  to a  $(\sigma \cup \{E\})$ -structure  $A'$  by setting

$$E(A') := \{(w(i), w(i+1)) \mid i \in [\ell-1]\} \cup \{(w(\ell), w(1))\}.$$

Then  $G'$  is the Gaifman graph of  $A'$ , which by Lemma 7.3.1 excludes  $K_c$  as a topological subgraph.

Using Lemma 7.1.2 we compute, for a suitable  $\tau \supseteq \sigma$ , a  $\tau$ -expansion  $A''$  of  $A'$  and an  $\text{FO}(\tau)$ -formula  $\varphi_{\text{succ}}^{(k)}(x, y)$  which defines a successor relation on  $A''$ . We replace all atomic subformulae  $\text{succ } xy$  in  $\varphi$  by  $\varphi_{\text{succ}}^{(k)}(x, y)$ , obtaining an  $\text{FO}(\tau)$ -formula  $\tilde{\varphi}$  such that

$$A'' \models \tilde{\varphi} \quad \Leftrightarrow \quad (A, S) \models \varphi$$

where  $S$  the successor relation defined by  $\varphi_{\text{succ}}^{(k)}$ . Note  $\varphi_{\text{succ}}^{(k)}$  and  $\tau$  depend only on  $k$ , which in turn only depends on  $H$ .

Since the Gaifman graph  $G''$  of  $A''$  excludes  $H$  as a topological subgraph, there is a class  $\mathcal{C}$  of graphs of bounded expansion such that  $G'' \in \mathcal{C}$ . We can therefore use Dvořák et al.'s model-checking algorithm [DKT10] for  $\text{FO}$  on  $\mathcal{C}$  to check whether

$$A'' \models \tilde{\varphi}$$

in time linear in  $|A|$ . □

## Chapter 8

# Dense Graphs

Most algorithmic meta-theorems have been obtained for sparse graph classes. This is because most techniques used in the design of these algorithms are robust under removing edges, yielding algorithms that work on monotone graph classes. For monotone graph classes however, we already mentioned in Chapter 6 that model checking for first-order logic is unlikely to be fixed-parameter tractable on any non-sparse class of graphs.

Among the few results known for dense graphs there are:

- On classes of graphs with bounded cliquewidth (or, equivalently, bounded rankwidth; cf. [OS06]), model checking even for monadic second-order logic has been shown to be fpt by Courcelle et al. [CMR00].
- More recently, model-checking on coloured posets of bounded width has been shown to be in fpt for existential FO by Bova et al. [BGS15] and for all of FO by Gajarský et al. [GHL<sup>+</sup>15].
- In [GHO<sup>+</sup>16], Gajarský et al. gave a structural characterisation of graph classes which are FO-interpretable in graphs of bounded degree, implying fpt model checking algorithms both for FO and succ-inv-FO on these graphs.
- On *map graphs*, a generalisation of planar graphs, we obtained an fpt model algorithm in [EK17].

The first two of these results extend to order-invariant FO, and therefore also to successor-invariant FO. For bounded cliquewidth, this has already been shown by Engemann et al. in [EKS12, Thm. 4.2], see also [EvdHK<sup>+</sup>19].

For posets of bounded width we gave a proof in [EK16], which we review below.

## Posets of Bounded Width

We first review the necessary definitions:

**Definition 8.0.1.** A *partially ordered set (poset)*  $(P, \leq^P)$  is a set  $P$  with a reflexive, transitive and antisymmetric binary relation  $\leq^P$ . A *chain*  $C \subseteq P$  is a totally ordered subset, i.e. for all  $x, y \in C$  one of  $x \leq^P y$  and  $y \leq^P x$  holds. An *antichain* is a set  $A \subseteq P$  such that if  $x \leq^P y$  for  $x, y \in A$  then  $x = y$ . The *width* of  $(P, \leq^P)$  is the maximal size  $|A|$  of an antichain  $A \subseteq P$ . A *coloured poset* is a poset  $(P, \leq^P)$  together with a function  $\lambda : P \rightarrow \Lambda$  mapping  $P$  to some set  $\Lambda$  of *colours*. By  $\|P\|$  we denote the length of a suitable encoding of  $(P, \leq^P)$ .

We will need Dilworth's Theorem, which relates the width of a poset to the minimum number of chains needed to cover the poset:

**Theorem 8.0.2** (Dilworth's Theorem). *Let  $(P, \leq^P)$  be a poset. Then the width of  $(P, \leq^P)$  is equal to the minimum number  $k$  of disjoint chains*

$$C_1, \dots, C_k \subseteq P$$

*needed to cover  $P$ , i.e. such that  $\bigcup_i C_i = P$ .*

A proof can be found in [Die12, Sec. 2.5]. Moreover, by a result of Felsner et al. [FRS03], both the width  $w$  and a set of chains  $C_1, \dots, C_w$  covering  $P$  can be computed from  $(P, \leq_P)$  in time  $O(w \cdot \|P\|)$ .

With this, we are ready to prove the following:

**Theorem 8.0.3.** *There is an algorithm which, on input a coloured poset  $(P, \leq^P)$  with colouring  $\lambda : P \rightarrow \Lambda$  and an order-invariant first-order formula  $\varphi$ , checks whether  $P \models \varphi$  in time  $f(w, |\varphi|) \cdot \|P\|^2$  where  $w$  is the width of  $(P, \leq^P)$ .*

*Proof.* Using the algorithm of [FRS03], we compute a chain cover  $C_1, \dots, C_w$  of  $(P, \leq^P)$ . To obtain a linear order on  $P$ , we just need to arrange the chains in a suitable order, which can be done by colouring the vertices with colours  $\Lambda \times [w]$  via

$$\lambda'(v) = (\lambda(v), j) \text{ for } v \in C_j.$$

Then

$$\varphi_{\leq}(x, y) := \left( \bigvee_{\substack{\lambda_x, \lambda_y \in \Lambda, \\ i < j}} (\lambda'(x) = (\lambda_x, i) \wedge \lambda'(y) = (\lambda_y, j)) \right) \vee \\ \left( \bigvee_{\substack{\lambda_x, \lambda_y \in \Lambda, \\ i \in [w]}} \lambda'(x) = (\lambda_x, i) \wedge \lambda'(y) = (\lambda_y, i) \wedge x \leq y \right)$$

defines a linear order on  $(P, \leq_P)$  with colouring  $\lambda'$ . After substituting  $\varphi_{\leq}$  for  $\leq$  in  $\varphi$  we may apply Gajarský et al.'s algorithm [GHL<sup>+</sup>15] to check whether  $P \models \varphi$ .  $\square$





# Bibliography

- [AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- [AHU74] Alfred V Aho, John E Hopcroft, and Jeffrey D Ullman. *The Design and Analysis of Computer Algorithms*. 1974.
- [Arm83] Mark Anthony Armstrong. *Basic Topology*. Springer, 1983.
- [AvMSS12] Matthew Anderson, Dieter van Melkebeek, Nicole Schweikardt, and Luc Segoufin. Locality from circuit lower bounds. *SIAM J. Comput.*, 41(6):1481–1523, 2012.
- [BDK12] Adam Bouland, Anuj Dawar, and Eryk Kopczynski. On tractable parameterizations of graph isomorphism. In *Proc. IPEC 2012*, pages 218–230, 2012.
- [BGS99] Andreas Blass, Yuri Gurevich, and Saharon Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1-3):141–187, 1999.
- [BGS15] Simone Bova, Robert Ganian, and Stefan Szeider. Model checking existential logic on partially ordered sets. *ACM Transactions on Computational Logic (TOCL)*, 17(2):10, 2015.
- [BIS90] David A Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within nc1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990.
- [Blu10] Achim Blumensath. Guarded second-order logic, spanning trees, and network flows. *Logical Methods in Computer Science*, 6(1:4):1–22, 2010.

- [Bod96] Hans L Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on computing*, 25(6):1305–1317, 1996.
- [BP16] Mikołaj Bojańczyk and Michał Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2016.
- [BP17] Mikołaj Bojańczyk and Michał Pilipczuk. Optimizing tree decompositions in mso. In *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- [BS05] Michael Benedikt and Luc Segoufin. Regular tree languages definable in fo. In *STACS*, pages 327–339, 2005.
- [BS09] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 74(1):168–186, 2009.
- [Cam94] Peter J. Cameron. *Combinatorics*. Cambridge University Press, 1994.
- [CH82] Ashok Chandra and David Harel. Structure and complexity of relational queries. *JCSS*, 25(1):pp. 99–128, 1982.
- [CK90] C.C. Chang and H.J. Keisler. *Model Theory*, volume 73 of *Studies in Logic*. North-Holland, 1990.
- [CMR00] Bruno Courcelle, Johann Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.
- [Cou90] Bruno Courcelle. Graph rewriting: An algebraic and logic approach. In *Formal Models and Semantics*, pages 193–242. Elsevier, 1990.
- [Cou91] Bruno Courcelle. The monadic second-order logic of graphs v: on closing the gap between definability and recognizability. *Theoretical Computer Science*, 80:pp. 153–202, 1991.

- [Cou96] Bruno Courcelle. The monadic second-order logic of graphs x: linear orderings. *Theoretical Computer Science*, 160(1–2):pp. 87–143, 1996.
- [Cou03] Bruno Courcelle. The monadic second-order logic of graphs xiv: Uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299(1-3):1–36, 2003.
- [DF99] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer, 1999.
- [DGHL09] Anuj Dawar, Martin Grohe, Bjarki Holm, and Bastian Laubner. Logics with rank operators. In *2009 24th Annual IEEE Symposium on Logic In Computer Science*, pages 113–122. IEEE, 2009.
- [DHK05] Erik D. Demaine, Mohammad Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation and coloring. In *FOCS*, 2005.
- [Die12] Reinhard Diestel. *Graph Theory*. Number 173 in GTM. Springer, 4th edition, 2012.
- [DKT10] Zdeněk Dvořák, Daniel Král, and Robin Thomas. Deciding first-order properties for sparse graphs. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010.
- [DLM07] Arnaud Durand, Clemens Lautemann, and Malika More. A simple proof of the polylog counting ability of first-order logic: Guest column. *SIGACT News*, 38(4):40–45, 12 2007.
- [DRR08] Anuj Dawar, David Richerby, and Benjamin Rossman. Choiceless polynomial time, counting and the cai–fürer–immerman graphs. *Annals of Pure and Applied Logic*, 152(1-3):31–50, 2008.
- [EEH14] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, pages 256–266, 2014.

- [EEH17] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Succinctness of order-invariant logics on depth-bounded structures. *ACM Transactions on Computational Logic (TOCL)*, 18(4):33, 2017.
- [EF99] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 2nd edition, 1999.
- [EFG16] Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. Order invariance on decomposable structures. In *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2016.
- [EFT94] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic*. Springer, 2nd edition, 1994.
- [EGT16] Michael Elberfeld, Martin Grohe, and Till Tantau. Where first-order and monadic second-order logic coincide. *ACM Transactions on Computational Logic (TOCL)*, 17(4):25, 2016.
- [EK16] Kord Eickmeyer and Ken-Ichi Kawarabayashi. Successor-invariant first-order logic on graphs with excluded topological subgraphs. In *Proceedings of the 25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, LIPIcs–Leibniz International Proceedings in Informatics, pages 18:1–18:15, 2016.
- [EK17] Kord Eickmeyer and Ken-ichi Kawarabayashi. Fo model checking on map graphs. In *International Symposium on Fundamentals of Computation Theory*, pages 204–216. Springer, 2017.
- [EKK13] Kord Eickmeyer, Ken-Ichi Kawarabayashi, and Stephan Kreutzer. Model checking for successor-invariant first-order logic on minor-closed graph classes. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '13*, pages 134–142. IEEE Computer Society, 2013.
- [EKS12] Viktor Engemann, Stephan Kreutzer, and Sebastian Siebertz. First-order and monadic second-order model-checking on or-

- dered structures. In *Logics in Computer Science*, pages 275–284, 2012.
- [EvdHK<sup>+</sup>19] Kord Eickmeyer, Jan van den Heuvel, Ken-ichi Kawarabayashi, Stephan Kreutzer, Patrice Ossona de Mendez, Michał Pilipczuk, Daniel Adolfo Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking on ordered structures. *Transactions on Computational Logic (submitted, recommended for publications with minor revisions)*, 2019.
- [FG01] Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM (JACM)*, 48(6):1184–1206, 2001.
- [FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [FRS03] Stefan Felsner, Vijay Raghavan, and Jeremy Spinrad. Recognition algorithms for orders of small width and graphs of small dilworth number. *Order*, 20(4):351–364, 2003.
- [Gai82] Haim Gaifman. On local and non-local properties. *Studies in Logic and the Foundations of Mathematics*, 107:105–135, 1982.
- [GHL<sup>+</sup>15] Jakub Gajarský, Petr Hliněný, Daniel Lokshtanov, Jan Obdržálek, Sebastian Ordyniak, MS Ramanujan, and Saket Saurabh. Fo model checking on posets of bounded width. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 963–974. IEEE, 2015.
- [GHO<sup>+</sup>16] J. Gajarský, P. Hliněný, J. Obdržálek, D. Lokshtanov, and M. S. Ramanujan. A new perspective on fo model checking of dense graph classes. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS ’16*, pages 176–184. ACM, 2016.
- [GKR13] Martin Grohe, Ken-ichi Kawarabayashi, and Bruce Reed. A simple algorithm for the graph minor decomposition. In *SODA*, 2013.
- [GKS17] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM (JACM)*, 64(3):17, 2017.

- [GM15] Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing*, 44(1):114–159, 2015.
- [GP15] Erich Grädel and Wied Pakusa. Rank logic is dead, long live rank logic! In *24th EACSL Annual Conference on Computer Science Logic*, page 390, 2015.
- [GR94] Zhicheng Gao and R. Bruce Richter. 2-walks in circuit graphs. *J. Comb. Theory, Ser. B*, 62(2):259–267, 1994.
- [GR08] Tobias Ganzow and Sasha Rubin. Order-invariant mso is stronger than counting mso in the finite. In *25th International Symposium on Theoretical Aspects of Computer Science*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.
- [Gro03] Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003.
- [Gro08] Martin Grohe. The quest for a logic capturing ptime. In *2008 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 267–271. IEEE, 2008.
- [GS00] Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Transactions on Computational Logic (TOCL)*, 1(1):112–130, 2000.
- [GS05] Martin Grohe and Nicole Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1:6):pp. 1–25, 2005.
- [HLN99] Lauri Hella, Leonid Libkin, and Juha Nurmonen. Notions of locality and their logical characterizations over finite models. *The Journal of Symbolic Logic*, 64(4):1751–1773, 1999.
- [How76] John M. Howie. *An introduction to semigroup theory*. Academic Press, 1976.
- [Hå86] John Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. Citeseer, 1986.

- [Imm82] Neil Immerman. Relational queries computable in polynomial time. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 147–152, 1982.
- [Imm99] Neil Immerman. *Descriptive Complexity*. Springer Verlag, 1999.
- [JW13] Gwenaél Joret and David R. Wood. Complete graph minors and the graph minor structure theorem. *Journal of Combinatorial Theory, Series B*, 103(1):61–74, 2013.
- [Kre11] Stephan Kreutzer. Algorithmic meta-theorems. In Javier Esparza, Christian Michaux, and Charles Steinhorn, editors, *Finite and Algorithmic Model Theory*, London Mathematical Society Lecture Note Series, chapter 5, pages 177–270. Cambridge University Press, 2011. a preliminary version is available at Electronic Colloquium on Computational Complexity (ECCC), TR09-147, <http://www.eccc.univ-trier.de/report/2009/147>.
- [Lib04] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Springer, 2004.
- [Liv82] A.B. Livchak. Languages for polynomial-time queries. In *Computer-based Modeling and Optimization of Heat-power and Electrochemical Objects*, page 41, 1982.
- [Lyn82] James F. Lynch. On sets of relations definable by addition. *Journal of Symbolic Logic*, 47(3):659—668, 1982.
- [Mak04] Johann A. Makowsky. Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic*, 126(1–3):pp. 159–213, 2004.
- [Min67] Marvin Lee Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.
- [MM63] John Moon and Leo Moser. Simple paths on polyhedra. *Pacific Journal of Mathematics*, 13(2):629–631, 1963.
- [MT01] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, 2001.



- [NOdM12] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*. Springer-Verlag Berlin Heidelberg, 2012.
- [OS06] Sang-Il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96:514 – 528, 2006.
- [Ott00] Martin Otto. Epsilon-logic is more expressive than first-order logic over finite structures. *The Journal of Symbolic Logic*, 65(4):1749–1757, 2000.
- [Pap03] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.
- [Pot94] Andreas Potthoff. *Logische Klassifizierung regulärer Baum-sprachen*. PhD thesis, Christian-Albrechts-Universität Kiel, 1994.
- [Ros07] Benjamin Rossman. Successor-invariant first-order logic on finite structures. *The Journal of Symbolic Logic*, 72(2):601–618, 2007.
- [RS03] Neil Robertson and Paul D Seymour. Graph minors. xvi. excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003.
- [Sch13] Nicole Schweikardt. A short tutorial on order-invariant first-order logic. In *Proc. CSR 2013*, pages 112–126, 2013.
- [See96] Detlef Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.
- [SS10] Nicole Schweikardt and Luc Segoufin. Addition-invariant fo and regularity. In *2010 25th Annual IEEE Symposium on Logic in Computer Science*, pages 273–282. IEEE, 2010.
- [Str94] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Birkhäuser, 1994.
- [Tho83] Carsten Thomassen. A theorem on paths in planar graphs. *Journal of Graph Theory*, 7(2):169–176, 1983.

- [Tho90] Carsten Thomassen. Embeddings of graphs with no short non-contractible cycles. *J. Comb. Theory, Series B*, 48:155–177, 1990.
- [Tut56] William T. Tutte. A theorem on planar graphs. *Transactions of the American Mathematical Society*, 82(1):99–116, 1956.
- [Var82] Moshe Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 137–146, 1982.
- [vdHKP<sup>+</sup>17] Jan van den Heuvel, Stephan Kreutzer, Michal Pilipczuk, Daniel A Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–11. IEEE, 2017.



# Index

- $2^X$ , 13
- $A \sqcup B$ , 77
- $A|_\sigma$ , 15
- $G[X]$ , 36
- $K[M]$ , 36
- $K_k$ , 36
- $L_q$ , 19, 21
- $N_A^r(a)$ , 59
- $[\cdot]_{\mathfrak{J}}$ , 16
- $[m, n]$ , 13
- $[n]$ , 13
- CMSO, 21
- $d\text{-exp}(n)$ , 13
- $\text{FO}(\sigma)$ , 17
- $\text{FO}_q$ , 19
- $\text{Fin}(\sigma)$ , 14
- $\text{Fin}_\sigma^{\text{conn}}$ , 40
- $\text{Graph}_C$ , 37
- $\sigma^<$ , 27
- $\sigma^{\text{succ}}$ , 29
- $\text{TD}_d^{\text{conn}}(\sigma)$ , 40
- $\text{TD}_d(\sigma)$ , 40
- $\alpha(A, a)$ , 63
- $\alpha_t$ , 102
- $\alpha_A$ , 64
- $\text{ar}(R)$ , 14
- $\beta \frac{a}{x}$ , 16
- $\binom{X}{k}$ , 13
- $\mathcal{A}_\sigma$ , 16
- $\mathcal{V}_{s \vdash t}$ , 39
- $\epsilon\text{-FO}$ , 28
- $\epsilon_x$ , 28
- $\equiv$ , 18
- $\equiv^C$ , 18
- $\exists^k \pmod{p}$ , 19
- $\mathfrak{J}$ , 16
- $\mathfrak{J} \frac{a}{x}$ , 16
- $\text{free}(\varphi)$ , 17
- $\gamma$ , 45
- $\equiv_q^L$ , 22
- $\mathcal{T}_{s \vdash t}$ , 39
- Graph**, 36
- $\equiv_{\wedge t}$ , 70
- $\models$ , 18
- $\models_R$ , 25
- $\equiv_{\text{mod } k}$ , 13
- $d\text{-EXP}$ , 14
- $<\text{-inv-FO}$ , 27
- $<\text{-inv-MSO}$ , 27
- $\partial X$ , 45
- $\preceq_1 \oplus \preceq_2$ , 23
- $\preceq_{\text{atomic}}$ , 65
- $\preceq$ , 47
- $\preceq_{L,q}$ , 65
- $\text{reach}_d(x, y)$ , 41
- $\text{rtp}_q^<(A)$ , 67
- $\text{succ-inv-FO}$ , 29
- $\sigma$ , 14

- $\sigma_G$ , 36
- $\sigma_{G,C}$ , 37
- $\simeq$ , 14
- $\text{td}(G)$ , 40
- $\text{FOmod}$ , 19
- $\text{MSO}$ , 20
- $\text{MSO}_q$ , 21
- $\preceq_{\text{top}}$ , 48
- $\text{tp}_q^<(A)$ , 67
- $k$ -walk, 92
  
- adhesion, 38
- adhesion set, 102
- ancestor, 37
- apex, 47
- arc, 45
- atomic type, 63
  
- bag, 38
  
- child, 37
- class, 15
  - elementary, 18
- colour, 37
- connected, 36
  - on a surface, 45
- contraction, 47
- cuff, 47
- curve, 45
  - polygonal, 45
  
- degenerate, 39
- drawing, 46
  - cellular, 46
  
- embeddable, 46
  - $s$ -nearly, 47
- endpoints, 35
- expansion, 15
  - ordered, 27
  
- successor, 29
  
- face, 46
- first-order logic, 17
- fixed-parameter tractable, 7, 86
- formula
  - atomic, 17
  - local, 60
- fpt, 7, 86
- frontier, 45
  
- Gaifman graph, 15
- graph, 35
  - $d$ -degenerate, 39
  - complete, 36
  - complete bipartite, 36
  - connected, 36
  - Gaifman, 15
  - planar, 46
  
- height (of a tree), 37
  
- image, 47
- internal vertex, 35
- internally vertex-disjoint, 35
- interpretation, 16
- isomorphic, 14
- isomorphism, 15
  
- leaf, 37
- logic
  - first-order, 17
  - guarded second-order, 21
  - monadic second-order, 20
- loop, 45
- $\text{L}_q$ -type, 22
  
- minor, 47
  - topological, 48
- minor image, 47

- model, 18
- model checking, 85
- modulo counting, 19, 21
- monadic second-order logic, 20
  
- negation normal form, 19
- neighbourhood, 59
- node, 38
  
- parent, 37
- path, 35
- path decomposition, 38
  
- $q$ -order, 66
- $q$ -type, 22
- quantifier alternation depth, 19
- quantifier rank, 18
- query, 15
  
- $\mathcal{R}$ -inv-L, 26
- $\mathcal{R}$ -invariant, 25
- reduct, 15
- region, 45
- root
  - treedepth, 40
- root (of a tree), 37
  
- sentence, 17
  - basic local, 60
- separation, 36
- sibling, 37
  
- signature, 14
- structure, 14
- subgraph, 36
  - induced, 36
  - topological, 48
- surface, 44
  
- term, 16
- theorem
  - Gaifman, 60
  - Graph Structure, 47
  - Jordan Curve, 45
- torso, 38
- tree, 37
  - ranked, 37
  - rooted, 37
  - sibbling, 37
- tree decomposition, 38
- treedepth, 40
- treewidth, 39
- type
  - $q$ -order, 67
  - $q$ -order root  $\sim$ , 67
  - atomic, 63
  
- universe, 14
  
- variable binding, 16
- vortex, 47
  
- width, 39